Smart Programming : YouTube Channel

An investment in Knowledge pays the best interest….

Smart Programming
We Educate - We Develop

☎ +91 62838-30308
Call us to Learn
Latest Technologies

City : Mohali (Punjab), & Chandigarh (India)

WEBSITE : http://www.smartprogramming.in

BUY COURSES ON : https://courses.smartprogramming.in

YOUTUBE CHANNEL : Smart Programming (https://www.youtube.com/c/SmartProgramming)

ANDROID APP : Smart Programming
(https://play.google.com/store/apps/details?id=com.smartprogramming)

https://www.facebook.com/smartprogramming.india

https://www.instagram.com/smart_programming

Java
android
python
Data Science
Machine Learning
Artificial Intelligence
M E A N
Full Stack
BIG DATA
hadoop

Java Introduction

=> What is Java ?

1. Programming Language : Java is a HIGH LEVEL OBJECT ORIENTED PROGRAMMING LANGUAGE.

2. Platform : Java is a platform because java has its own runtime environment (JRE)

=> Editions of Java :-

1. J2SE : -> Java 2 Standard Edition (Core Java)

-> In this edition we learn java fundamentals

-> Projects : Standalone Applications

2. J2EE : -> Java 2 Enterprise Edition (Advance Java)

-> In this edition we learn server side programming

-> Projects : Enterprise Applications

3. J2ME : -> Java 2 Micro Edition (it has been replaced by Android)

-> In this edition we learn micro programming

-> Projects : Mobile Based Applications

=> Java History :-

-> Company : Sun Microsystems (Oracle)

-> Designed By : James Gosling & his team

-> Idea : Java idea came because designers wanted to create a language which should be simple, platform independent, tight coded programming language etc

-> Name : Oak (Java)

-> Year : 1991 (development) -> 1995 (release)

-> Symbol : Coffee

=> Versions of Java :-

-> https://en.wikipedia.org/wiki/Java_version_history

=> What is difference between C, C++ & Java :-

1. PROGRAMMING PARADIGM :

-> C : Procedural Language

-> C++ : OOP (Object Oriented Programming)

-> Java : Pure OOP (Object Oriented Programming) - 99%

2. ORIGIN :-

-> C : Based on Assembly Language

-> C++ : Based on C Language

-> Java : Based on C & C++

3. DESIGNED BY/YEAR :-

-> C : Dennis Ritchie / 1972

-> C++ : Bjarne Stroustrup / 1979

-> Java : James Gosling / 1995

4. TRANSLATOR :-

-> C : Compiler

-> C++ : Compiler

-> Java : Interpreter & Compiler

5. FILE GENERATED :-

-> C : .exe file

-> C++ : .exe file

-> Java : .class file

6. STATIC/DYNAMIC PROGRAMMING LANGUAGE :

-> C : Static Programming Language

-> C++ : Static Programming Language

-> Java : Dynamic Programming Language

7. PRE-PROCESSOR DIRECTIVES :-

-> C : Support header files (#include, #define)

-> C++ : Support header files (#include, #define)

-> Java : Use packages (import)

8. PLATFORM DEPENDENT/INDEPENDENT :-

-> C : Platform Dependent

-> C++ : Platform Dependent

-> Java : Platform Independent

9. POINTERS :-

-> C : Support pointers

-> C++ : Support pointers

-> Java : Does not support pointers

10. INHERITANCE :-

-> C : No inheritance

-> C++ : Supports inheritance

-> Java : Supports inheritance but not multiple inheritance

11. OVERLOADING :-

-> C : Does not support overloading

-> C++ : Supports overloading

-> Java : Supports overloading but not operator overloading

12. DESTRUCTORS :-

-> C : Does not support constructor and destructor

-> C++ : Supports destructor

-> Java : Does not support destructor

13. CALL BY VALUE / CALL BY REFERENCE :-

-> C : Supports both

-> C++ : Supports both

-> Java : Support only Call By Value

Interview Questions :-

=> What is static or dynamic programming language?

=> How java is dynamic programming language ?

=> Why java does not require pre-processor ?

=> How java is platform independent ?

=> Why java does not support pointers ?

=> What is call by reference and call by value ?

=> Why java has good memory management as compared to C & C++ ?

Company Links & Contacts

Company Name: Smart Programming (+91 62838-30308)

Address : Chandigarh & Mohali (Punjab), India

Websites: https://www.smartprogramming.in/

https://courses.smartprogramming.in

Android App:
https://play.google.com/store/apps/details?id=com.smartprogramming

YouTube Channel:
https://www.youtube.com/c/SmartProgramming

Smart Programming : YouTube Channel

An investment in Knowledge pays the best interest….

Smart Programming
We Educate - We Develop

☎ +91 62838-30308
Call us to Learn
Latest Technologies

City : Mohali (Punjab), & Chandigarh (India)

WEBSITE : http://www.smartprogramming.in

BUY COURSES ON : https://courses.smartprogramming.in

YOUTUBE CHANNEL : Smart Programming (https://www.youtube.com/c/SmartProgramming)

ANDROID APP : Smart Programming
(https://play.google.com/store/apps/details?id=com.smartprogramming)

https://www.facebook.com/smartprogramming.india

https://www.instagram.com/smart_programming

Java
android
python
Data Science
Machine Learning
Artificial Intelligence
M E A N
Full Stack
BIG DATA
hadoop

Java/Python Introduction

=> What is difference between Java & Python ?

1. PROGRAMMING PARADIGM:-

-> Java : Pure OOP (Object Oriented Programming) (from JavaSE 8 version Java introduced some procedural oriented programming paradigm features like Lamba Expression, functional interface, Functions, Predicates, Module System (java9) etc)

-> Python : Procedural Oriented and OOP

2. STATICALLY OR DYNAMICALLY TYPED LANGUAGE:-

-> Java : Statically Typed Programming Language (int rollno=101; String name="deepak";)

-> Python : Dynamically Typed Programming Language (rollno=101;)

3. COMPILATIVE OR INTERPRETIVE LANGUAGE:-

-> Java : It is both compilative and interpretive programming language

-> Python : It is interpretive programming language

4. TRANSLATOR:-

-> Java : It has compiler and interpreter

-> Python : It has only interpreter

5. SPEED:-

-> Java : It is fast as compared to python

-> Python : It is slower as compared to java

6. PORTABILITY:-

-> Java : As java supports more hardwares like computers, mobiles, remotes, tv etc thus java is more portable

-> Python : It is less portable as comapred to java

7. INHERITANCE:-

-> Java : It supports except multiple inheritance

-> Python : It supports all inheritances

8. USE OF ";":-

-> Java : Its mendatory to use ; in java

-> Python : Its not mendatory to use ; in python

9. USE OF {}:-

-> Java : Its mendatory to use {} in java

-> Python : {} are not used in python (python used indentation)

10. SYNTAX:-

-> Java : Hard Coded (if you miss any ; or {} then it will provide an error)

-> Python : Easy Syntax (easy to remember as it is close to humans)

11. LENGTH OF CODE:-

-> Java : Java has more number of lines of code

-> Python : Python has less number of lines of code

12. DATABASE COMPITABILITY:-

-> Java : It has more compatibility with database as compared to python (JDBC - Java Database Connectivity)

-> Python : It is less compatible with database and due to this reason python is rarely used in distributed (enterprise) applications.

13. FRAMEWORKS:-

-> Java : Java provides more frameworks for eg Spring, Hibernate, JSF, Struts etc

-> Python : Python provides less framework as compared to java for eg Django, Flask etc

14. MACHINE LEARNING LIBRARIES:-

-> Java : Weka, Mallet, Deeplearning4j, MOA etc

-> Python : Tensorflow, Pythorch etc

15. BEST FOR WHICH AREA:-

-> Java : Java is best for Desktop GUI Applications, Enterprise Applications, Mobile Applications, Embded Systems etc

-> Python : Python is best for Scientic & Numeric Computing, Machine Learning, AI etc

---

=> Java Features :-

1. Simple :-

-> Java is using all the simplified syntax from C & C++

-> Some C & C++ features were either removed (pointers, multiple inheritance) from java or was created automatic (memory management)

-> Java has less execution time period and less maintenance (less memory, less power)

2. PlatForm Independent:-

-> Java is a platform independent language which means java can be executed on any machine or operating system

3. Portable:-

-> Java is portable language as it can be executed on multiple hardwares

4. OOP (Object Oriented Programming):-

-> Java is OOP language thus java works around objects

-> Java supports many OOP features like classes, objects and methods, inheritance, polymorphism, abstraction, encapsulation etc.

5. Security:-

-> Java is very secured language because

1. Java does not have pointers
2. Java has byte code verifier
3. Java has a security manager that defines the access for java classes

6. Robust:-

-> Java is Robust i.e. powerful language because

1. Java provides a lot of classes and interfaces to handle the exceptions
2. Automatic Memory Management

7. Multithreaded Language:-

-> Java supports multithreading

8. High Performance:-

-> Java has high performance because :-

1. Robust
2. Portable
3. Multithreaded Language

etc

9. Distributed Applications:-

-> Java can be used for distributed application (enterprise applications)

10. Java is compilative and interpretive language

Interview Questions:-

=> What do you mean by Platform Independent ?

=> How java is Platform Independent ?

=> Is JVM Platform Dependent or Platform Independent ?

Company Links & Contacts

Company Name: Smart Programming (+91 62838-30308)

Address : Chandigarh & Mohali (Punjab), India

Websites: https://www.smartprogramming.in/

https://courses.smartprogramming.in

Android App: https://play.google.com/store/apps/details?id=com.smartprogramming

YouTube Channel: https://www.youtube.com/c/SmartProgramming

Smart Programming : YouTube Channel

An investment in Knowledge pays the best interest….

Smart Programming
We Educate - We Develop

+91 62838-30308
Call us to Learn
Latest Technologies

City : Mohali (Punjab),
& Chandigarh
(India)

WEBSITE : http://www.smartprogramming.in

BUY COURSES ON : https://courses.smartprogramming.in

YOUTUBE CHANNEL : Smart Programming (https://www.youtube.com/c/SmartProgramming)

ANDROID APP : Smart Programming
(https://play.google.com/store/apps/details?id=com.smartprogramming)

https://www.facebook.com/smartprogramming.india

https://www.instagram.com/smart_programming

Java
android
python
Data Science
Machine Learning
Artificial Intelligence
M E A N
Full Stack
BIG DATA
hadoop

Java Softwares

Download & Install

Java Softwares :

1. JDK (Java Development Kit) :

Latest version : Java SE 15

LTS (Long Term Service) : Java SE 8, JavaSE11

2. Editor :

Notepad, Notepad++

3. IDE's (Integrated Development Environment) :

-> An integrated development environment is a software application that provides comprehensive facilities to computer programmers for software development. An IDE normally consists of at least a source code editor, build automation tools and a debugger.

-> IDE used for Java are :-

a. Eclipse (Java SE, Java EE) - 2020-09 | 16 September 2020 | 4.17

b. Netbeans (Java SE, Java EE) - Netbeans IDE 8.2 was released on 3 October 2016

Company Links & Contacts

Company Name: Smart Programming (+91 62838-30308)

Address : Chandigarh & Mohali (Punjab), India

Websites: https://www.smartprogramming.in/

https://courses.smartprogramming.in

Android App: https://play.google.com/store/apps/details?id=com.smartprogramming

YouTube Channel: https://www.youtube.com/c/SmartProgramming

Smart Programming : YouTube Channel

An investment in Knowledge pays the best interest….

Smart Programming
We Educate - We Develop

☎ +91 62838-30308
Call us to Learn Latest Technologies

City : Mohali (Punjab), & Chandigarh (India)

WEBSITE : http://www.smartprogramming.in

BUY COURSES ON : https://courses.smartprogramming.in

YOUTUBE CHANNEL : Smart Programming (https://www.youtube.com/c/SmartProgramming)

ANDROID APP : Smart Programming (https://play.google.com/store/apps/details?id=com.smartprogramming)

https://www.facebook.com/smartprogramming.india

https://www.instagram.com/smart_programming

Java
android
python
Data Science
Machine Learning
Artificial Intelligence
M E A N
Full Stack
BIG DATA
hadoop

Steps to Prepare

Java First Program

=> Steps to prepare Java First Program :-

1. Download & Install Java Softwares

   (JDK, Editor, IDE's)

2. Create Java First Program

   2.1 Java Naming Conventions

   2.2 Structure of Java Program

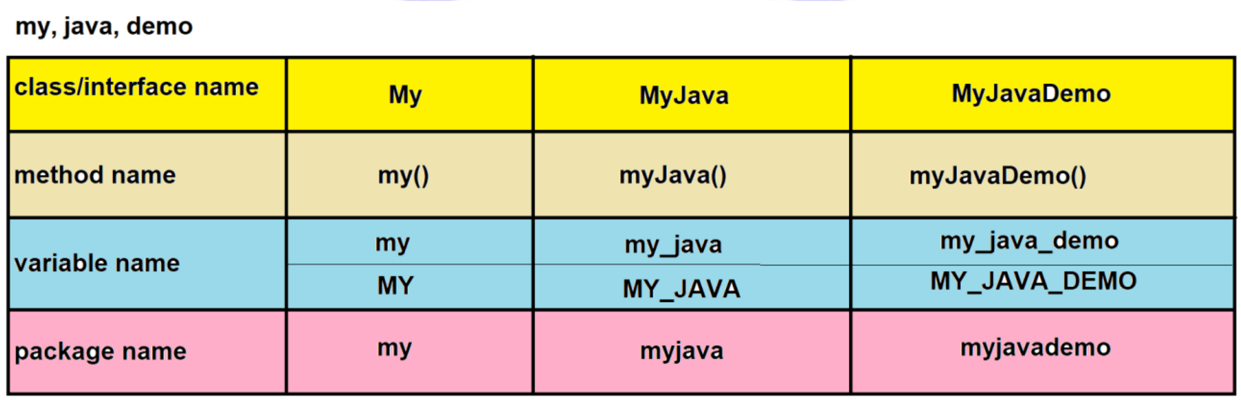
   2.3 Java Program Explanation

3. Compile & Run Java Program

=> Java Naming Conventions :-

-> Java is case sensitive language which means that declaring upper case and lower case name are different (int a=10; int A=10;)

-> How to declare names :-



my, java, demo

class/interface name	My	MyJava	MyJavaDemo
method name	my()	myJava()	myJavaDemo()
variable name	my MY	my_java MY_JAVA	my_java_demo MY_JAVA_DEMO
package name	my	myjava	myjavademo

=> Structure Of Java Program :-

-> There are 5 main sections in a java program :

1. Documentation Section
2. Package Statement
3. Import Statement
4. Interface/Class Section
    -> Variables & Methods

5. Main Class Section

-> main method

1. Documentation Section :-

-> It contains the basic information about java program for eg. author's name, date of creation, version, program name, company name, client name, description etc

-> This part is OPTIONAL

-> To create documentation section we use comments

-> There are 3 types of comments :-

1. Single Line Comment (//)

2. Multi-Line Comment (/* ----- */)

3. Documentation Comment (/**-*-*-*-*-*/)

-> To provide description or metadata of the program Java introduced new feature in JDK 1.5 version i.e. "Annotation" (@---)

-> If we use comments for documentation section then at compilation phase these comments will be removed by the compiler which cannot be read further, so to resolve this problem java introduced annotation which cannot be removed at compilation or runtime.

-> If you want to provide documentation section in html form, then we can use following command in CMD :-

```
javadoc FileName.java
```

-> javap -version (to get JDK version)

-> javap packagename.ClassName (to get the class details)

Company Links & Contacts

Company Name: Smart Programming (+91 62838-30308)

Address : Chandigarh & Mohali (Punjab), India

Websites: https://www.smartprogramming.in/

https://courses.smartprogramming.in

Android App:
https://play.google.com/store/apps/details?id=com.smartprogramming

YouTube Channel:
https://www.youtube.com/c/SmartProgramming

Smart Programming : YouTube Channel

An investment in Knowledge pays the best interest….

Smart Programming
We Educate - We Develop

+91 62838-30308
Call us to Learn
Latest Technologies

City : Mohali (Punjab),
& Chandigarh
(India)

WEBSITE : http://www.smartprogramming.in

BUY COURSES ON : https://courses.smartprogramming.in

YOUTUBE CHANNEL : Smart Programming (https://www.youtube.com/c/SmartProgramming)

ANDROID APP : Smart Programming
(https://play.google.com/store/apps/details?id=com.smartprogramming)

https://www.facebook.com/smartprogramming.india

https://www.instagram.com/smart_programming

Java
android
python
Data Science
Machine Learning
Artificial Intelligence
M E A N
Full Stack
BIG DATA
hadoop

Java Program Structure

& Explanation

=> Steps to prepare Java First Program :-

1. Download & Install Java Softwares
    (JDK, Editor, IDE's)
2. Create Java First Program
    - 2.1 Java Naming Conventions
    - 2.2 Structure of Java Program
    - 2.3 Java Program Explanation
3. Compile & Run Java Program

=> Structure Of Java Program :-

-> There are 5 main sections in a java program:-

1. Documentation Section
2. Package Statement
3. Import Statement

4. Interface/Class Section

    -> Variables & Methods

5. Main Class Section

    -> main method

1. Documentation Section :-

-> It contains the basic information about java program for eg. author's name, date of creation, version, program name, company name, client name, description etc

-> This part is OPTIONAL

-> To create documentation section we use comments

-> There are 3 types of comments :-

1. Single Line Comment (//)
2. Multi-Line Comment (/* ----- */)

3. Documentation Comment (/**-*-*-*-*-*/)

-> To provide description or metadata of the program Java introduced new feature in JDK 1.5 version i.e. "Annotation" (@---)

-> If we use comments for documentation section then at compilation phase these comments will be removed by the compiler which cannot be read further, so to resolve this problem java introduced annotation which cannot be removed at compilation or runtime.

-> If you want to provide documentation section in html form, then we can use following command in CMD :-

```
javadoc FileName.java
```

-> javap -version (to get JDK version)

-> javap packagename.ClassName (to get the class details)

2. Package Statement :-

-> It is a group of similar type of classes or interfaces or packages

-> Advantage of package :-

1. It prevents naming conflicts of classes or interfaces

2. Easy to maintain

3. Accessability to classes or interfaces can be controlled (Security)

4. Reusability

-> Types of packages :-

1. Predefined Package (which are already created by java for eg. lang, awt, swing, util, sql etc)

2. User Defined Package (which are created by the developer)

-> Syntax : package package_name;

-> Example :-

```
package techdept.aa.bb;
```

```
class Employee
        {


        }
```

-> Conditions for package :-

1. Package statement should be only one in the single java file

2. Package statement should be the first statement in java file

-> Java provides one facility for naming convention for package that is we can create package on the same name of our domain name but in reverse order. For eg.
com.smartprogramming.usajohn.techdept
com.smartprogramming.usajohn.managingdept

-> This is OPTIONAL but you should always put all the classes or interfaces in any one package

3. Import Statement :-

-> Import statements is used to make available the classes or interfaces which we are going to use in present java file

-> Syntax :

1. import packagename.subpackagename.ClassName;

2. import packagename.*;

-> if we use "import packagename.*;" only classes or interfaces will be imported, not the subpackage.

-> If we dont want to use import statement then we can directly write the packagename.ClassName to use that class.

-> This is OPTIONAL to use but mainly we use it in major level projects.

-> This line is written after the package line.

-> java.lang package is the only package which is imported by default

4. Interface/Class Section : -> Variables & Methods

-> In this section we create classes or interfaces which contains variables, methods, constructors, blocks etc.

5. Main Class Section -> main method :

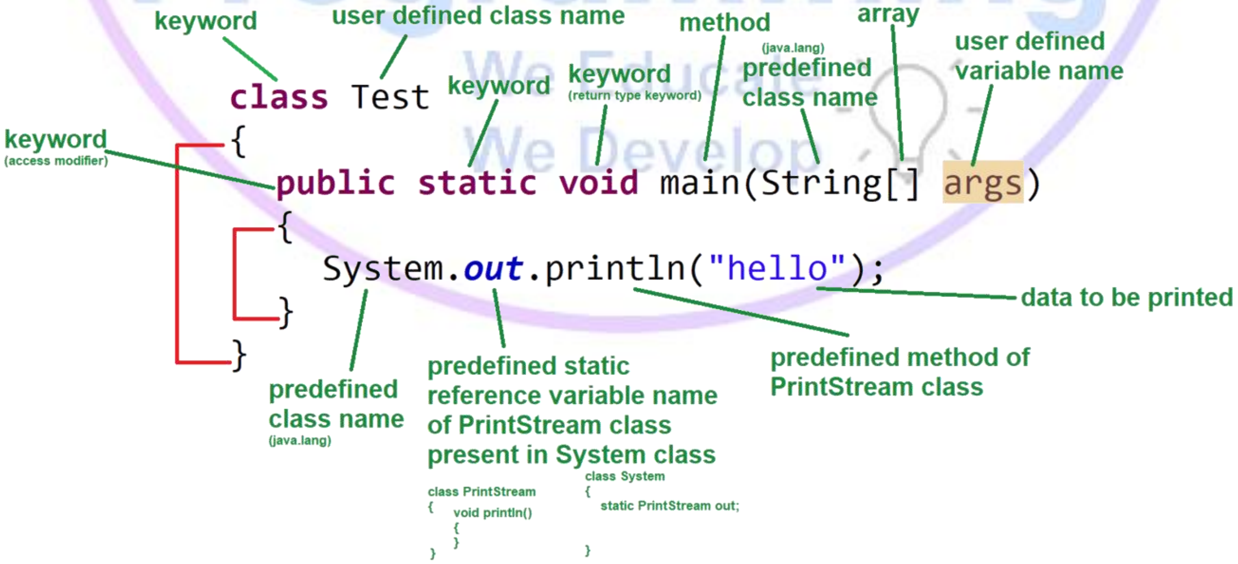
-> In every project or java file there should be one main class which should contain main() method from which the execution of the program will start.

-> Syntax :

```
class Test
{
        public static void main(String[] args)
        {


        }
}
```

=> Java Program Explanation :-

```
class Test
{
        public static void main(Stirng[] args)
        {
                System.out.println("hello");
        }
}
```



```
class Test
{
    public static void main(String[] args)
    {
        System.out.println("hello");
    }
}
```

keyword
user defined class name
keyword
keyword
(return type keyword)
method
(java.lang)
predefined
class name
array
user defined
variable name
keyword
(access modifier)
data to be printed
predefined
class name
(java.lang)
predefined static
reference variable name
of PrintStream class
present in System class
predefined method of
PrintStream class

```
class PrintStream
{       void println()
        {
        }
}
```

```
class System
{
    static PrintStream out;


}
```

Interview Questions :-

=> Explanation of each and every word in simple java hello program

Company Links & Contacts

Company Name: Smart Programming (+91 62838-30308)

Address : Chandigarh & Mohali (Punjab), India

Websites: https://www.smartprogramming.in/

https://courses.smartprogramming.in

Android App: https://play.google.com/store/apps/details?id=com.smartprogramming

YouTube Channel: https://www.youtube.com/c/SmartProgramming

Smart Programming : YouTube Channel

An investment in Knowledge pays the best interest….

Smart Programming

We Educate - We Develop

☎ +91 62838-30308

Call us to Learn Latest Technologies

City : Mohali (Punjab), & Chandigarh (India)

WEBSITE : http://www.smartprogramming.in

BUY COURSES ON : https://courses.smartprogramming.in

YOUTUBE CHANNEL : Smart Programming (https://www.youtube.com/c/SmartProgramming)

ANDROID APP : Smart Programming (https://play.google.com/store/apps/details?id=com.smartprogramming)

https://www.facebook.com/smartprogramming.india

https://www.instagram.com/smart_programming

Java
android
python
Data Science
Machine Learning
Artificial Intelligence
M E A N
Full Stack
BIG DATA
hadoop

Compile & Run Java Program

=> Steps to prepare Java First Program :-

1. Download & Install Java Softwares

   (JDK, Editor, IDE's)

2. Create Java First Program

   2.1 Java Naming Conventions

   2.2 Structure of Java Program

   2.3 Java Program Explanation

3. Compile & Run Java Program

---

3. Compile & Run Java Program :-

3.1 Open CMD

3.2 Type command javac and press enter (this step is to check whether java path is set or not) - if it says that "'javac' is not recognized as an internal or external command, operable program or batch file." that means java path is not set

3.3 If path is not set, then we have to set path and to set java path we have 2 options :-

-> Temporary path set

-> Permanent path set

3.4 Open the directory in CMD in which you have saved the java program

3.5 Compile the program (javac FileName.java)

3.6 Run the program (java MainClassName)

Cases :-

1. If file name and class name is not same

-> .class file will be created on the name of class that we have created in java file

-> Number of .class files that will be created depends on the number of classes that we have created in java file

-> If the class is public then we have to save the file name as that class name only, but if there is no public class, then we can save the file name by any name

-> We cannot create more than one public class in a single java file

2. main() method cases :-

-> public static void main(String[] args) - preferred way

-> static public void main(String[] args)

-> public static void main(String[   ] args)

-> public static void main(String args[])

-> public static void main(String xyz[])

-> synchronized final public static void main(String[] args)

-> public static void main(String... args)

-> public void static main(String[] args) - Incorrect because return type and method name should be in sequence or syntax

---

Interview Questions :-

=> How to set path in java

=> What is difference between temporary path set and permanent path set

=> What happens in java compilation phase

=> Different ways to declare main() method

Company Links & Contacts

Company Name: Smart Programming (+91 62838-30308)

Address : Chandigarh & Mohali (Punjab), India

Websites: https://www.smartprogramming.in/

https://courses.smartprogramming.in

Android App:
https://play.google.com/store/apps/details?id=com.smartprogramming

YouTube Channel:
https://www.youtube.com/c/SmartProgramming

Smart Programming : YouTube Channel

An investment in Knowledge pays the best interest….

Smart Programming
We Educate - We Develop

☎ +91 62838-30308
Call us to Learn
Latest Technologies

City : Mohali (Punjab),
& Chandigarh
(India)

WEBSITE : http://www.smartprogramming.in

BUY COURSES ON : https://courses.smartprogramming.in

YOUTUBE CHANNEL : Smart Programming (https://www.youtube.com/c/SmartProgramming)

ANDROID APP : Smart Programming
(https://play.google.com/store/apps/details?id=com.smartprogramming)

https://www.facebook.com/smartprogramming.india

https://www.instagram.com/smart_programming

Java
android
python
Data Science
Machine Learning
Artificial Intelligence
M E A N
Full Stack
BIG DATA
hadoop

main() Method Syntaxes

=> main() method syntaxes :-

-> public static void main(String[] args) - Correct

-> public static void main(Integer[] args) - Correct syntax for method but incorrect syntax for main() method

-> static public void main(String[] args) - Correct because we can change the sequene of static and public

-> public void static main(String[] args) - Incorrect & Compile time error because method syntax is incorrect as java needs always method name after return type

-> public static void args(String[] args) or public static void xyz(String[] args) - Correct syntax for method but incorrect for main() method

-> public static void main(String[   ] args) - Correct

-> public static void main(String   []   args) - Correct

-> public static void main(String   []args) - Correct

-> public static void main(String args[]) - Correct

-> public static void main(String args() {}) - Incorrect

-> public static void main(String args) - Correct syntax for method but incorrect syntax for main() method

-> public static void main(String[] xyz) - Correct

-> public static void main(String... xyz) - Correct (but there should be always 3 dots only)

-> protected/private static void main(String[] args) - Correct syntax for method but incorrect syntax for main() method

-> final public static void main(String[] args) - Correct

-> synchronized public static void main(String[] args) - Correct

-> strictfp public static void main(String[] args)- Correct

-> public void main(String[] args) - Correct syntax for method but incorrect syntax for main() method (Error : Main method is not static in class Test)

-> static void main(String[] args) - Correct syntax for method but incorrect syntax for main() method

Company Links & Contacts

Company Name: Smart Programming (+91 62838-30308)

Address : Chandigarh & Mohali (Punjab), India

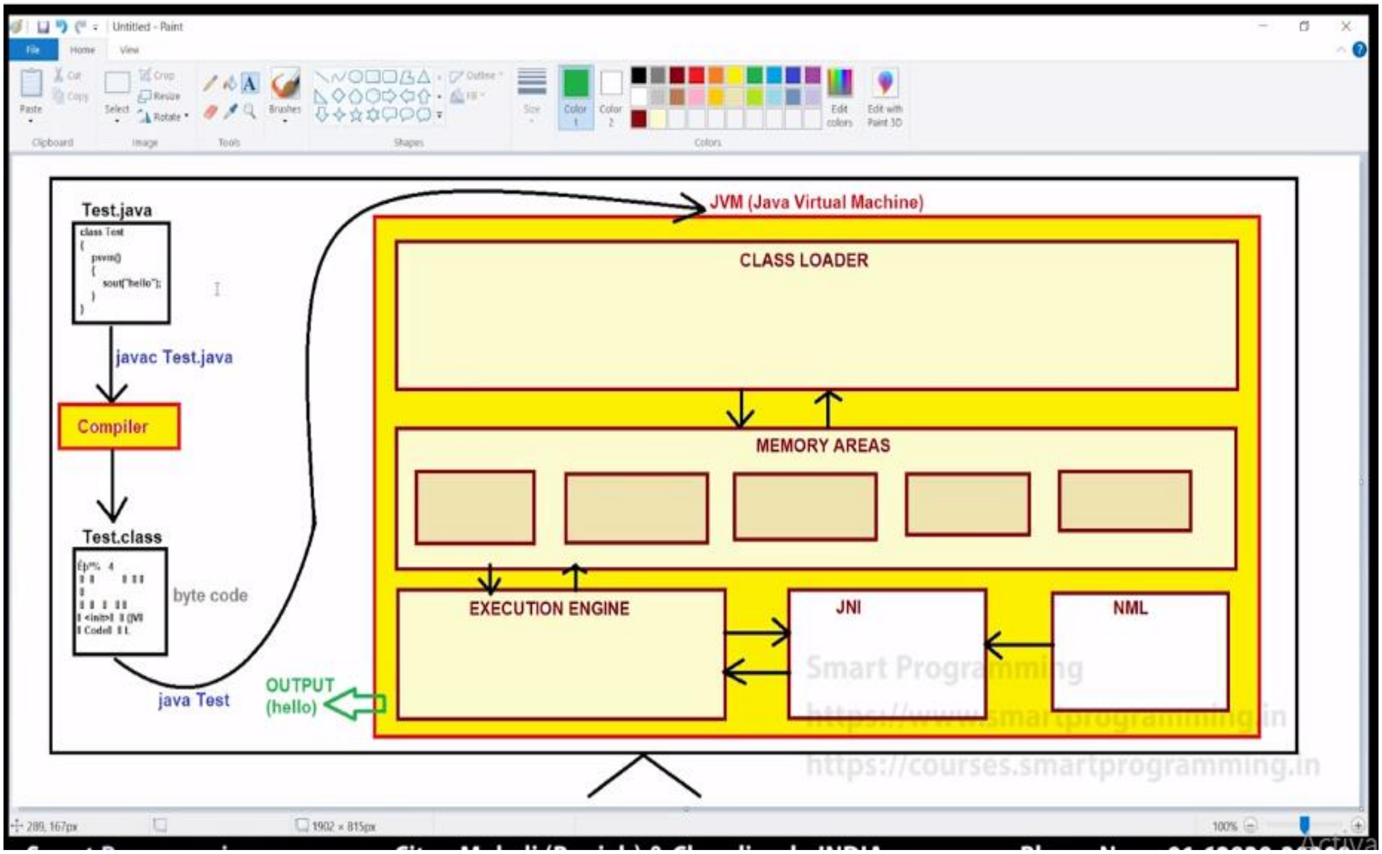
Websites: https://www.smartprogramming.in/

https://courses.smartprogramming.in

Android App:
https://play.google.com/store/apps/details?id=com.smartprogramming

YouTube Channel:
https://www.youtube.com/c/SmartProgramming

Deep Explanation of compilation & Execution of Java Programs:===


Untitled - Paint
File
Home
View
Test.java
javac Test.java
Compiler
Test.class
byte code
java Test
OUTPUT
(hello)
JVM (Java Virtual Machine)
CLASS LOADER
MEMORY AREAS
EXECUTION ENGINE
JNI
NML
Smart Programming
https://www.smartprogramming.in
https://courses.smartprogramming.in

```
class Test
{
  psvm()
  {
    sout("hello");
  }
}
```

Interview Questions:--

⇨ What is the role of Compiler? OR,

⇨ what happens in java compilation phase.

1. Compiler checks the syntax whether it is correct or not. (whether the class,method,interfaces,variable declaration etc. is correct or not, java is case sensitive language thus it checks upper case and lowercase keywords names.)

2. Java is strongly typed language thus it checks whether the variables has stored correct value or not.
   For example:--int a="Abhishek"; //now here Abhishek is string and not allowed to assign into int type variable a.
3. Compiler ignores all typed of comments.
4. Compiler generates the byte code.

- ⇨ What is the role of JVM?
- ⇨ What happens in java execution phase?

1. Byte code is loaded in JVM.
2. Byte code verifier checks whether the byte coe is correct or not.
3. Memory allocation starts (variables,class,information etc.)
4. Interpreter and JIT compiler executes our program and provides the output.

- ⇨ What is virtual machine?
  - ➔ It is software simulation of a machine performs operation similar to physical machine.
  - ➔ Types of virtual machine---
    - o Hardware based virtual machine or system based virtual machine.
    - o Application based virtual machine or process based virtual machine .
  - ➔ JVM is Application based or Process based VM.

- ⇨ What is JVM(Java Virtual Machine)?
  - ➔ JVM is Java Virtual Machine which is used to execute java byte code .
  - ➔ JVM is Application based or process based VM.
  - ➔ Role of JVM.
    - o Architechture of JVM:-> figure is given below at very last.

====================================

- ⇨ Explain deep architechture of Class loader
- ⇨ What data is stored in Method Area?

---.class file information

---static variables

- ⇨ How many memory areas are present in JVM?
  - ➔ Five types i) Method Area ,ii)Heap Area iii)Stack Area , iv) PC Register , v) Native Method Area
- ⇨ What data is stored in heap area?

---objects

----arrays

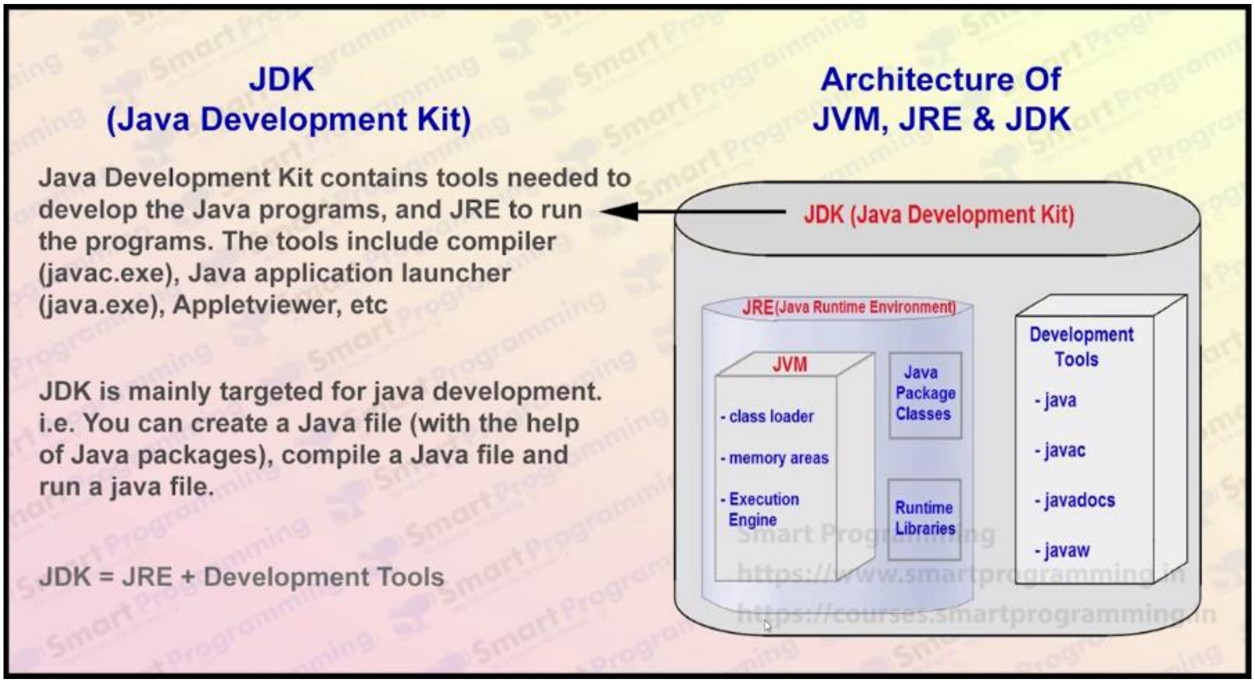
----instance variables

- ⇨ What data is stored in Stack Area?

--- current running method

----local variables

- ⇨ Explain the working of execution engine.
- ⇨ What is JIT compiler?



JDK
(Java Development Kit)

Java Development Kit contains tools needed to develop the Java programs, and JRE to run the programs. The tools include compiler (javac.exe), Java application launcher (java.exe), Appletviewer, etc

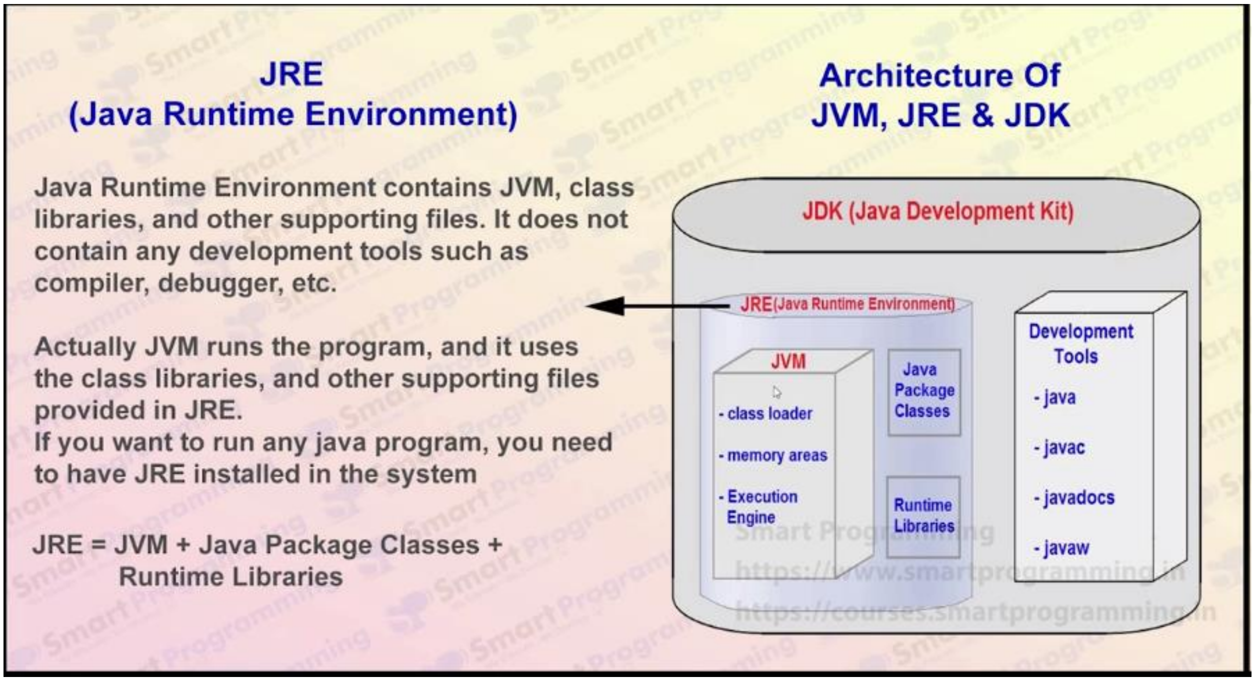
JDK is mainly targeted for java development. i.e. You can create a Java file (with the help of Java packages), compile a Java file and run a java file.

JDK = JRE + Development Tools

Architecture Of JVM, JRE & JDK

JDK (Java Development Kit)
JRE(Java Runtime Environment)
JVM
- class loader
- memory areas
- Execution Engine
Java Package Classes
Runtime Libraries
Development Tools
- java
- javac
- javadocs
- javaw

Smart Programming
https://www.smartprogramming.in
https://courses.smartprogramming.in



JRE
(Java Runtime Environment)

Java Runtime Environment contains JVM, class libraries, and other supporting files. It does not contain any development tools such as compiler, debugger, etc.

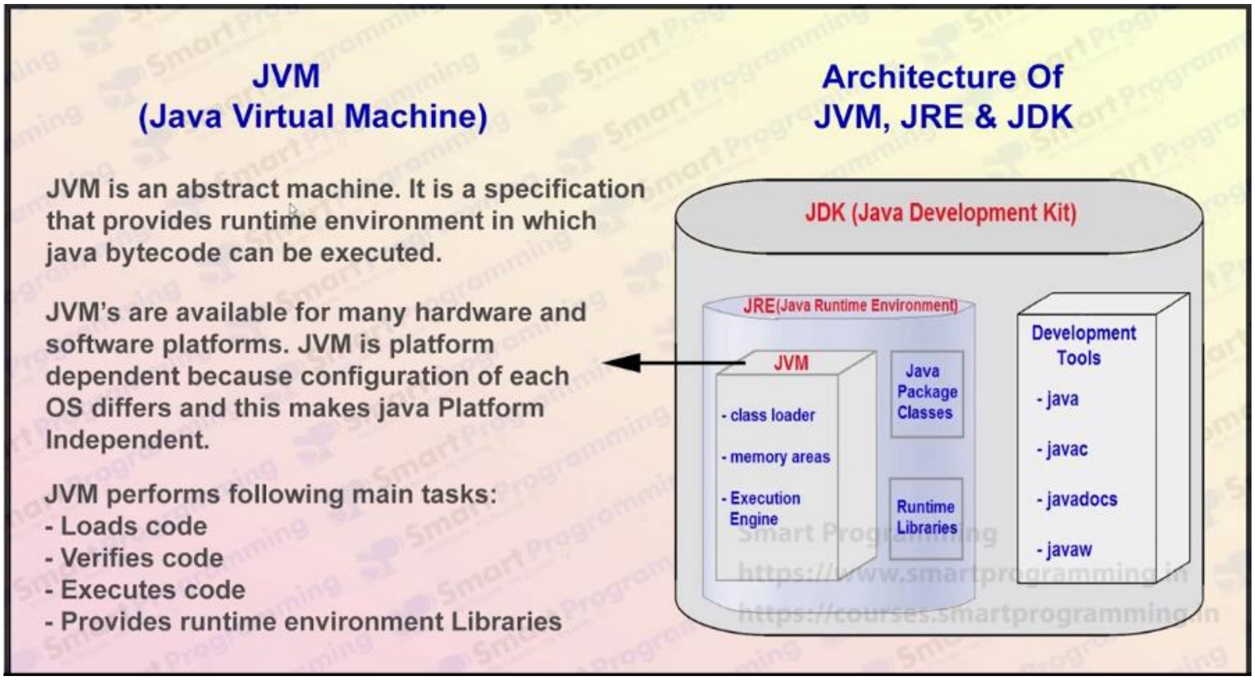
Actually JVM runs the program, and it uses the class libraries, and other supporting files provided in JRE.
If you want to run any java program, you need to have JRE installed in the system

JRE = JVM + Java Package Classes +
Runtime Libraries

Architecture Of JVM, JRE & JDK

JDK (Java Development Kit)
JRE(Java Runtime Environment)
JVM
- class loader
- memory areas
- Execution Engine
Java Package Classes
Runtime Libraries
Development Tools
- java
- javac
- javadocs
- javaw

Smart Programming
https://www.smartprogramming.in
https://courses.smartprogramming.in



JVM
(Java Virtual Machine)

JVM is an abstract machine. It is a specification that provides runtime environment in which java bytecode can be executed.

JVM's are available for many hardware and software platforms. JVM is platform dependent because configuration of each OS differs and this makes java Platform Independent.

JVM performs following main tasks:
- Loads code
- Verifies code
- Executes code
- Provides runtime environment Libraries

Architecture Of
JVM, JRE & JDK

JDK (Java Development Kit)

JRE(Java Runtime Environment)

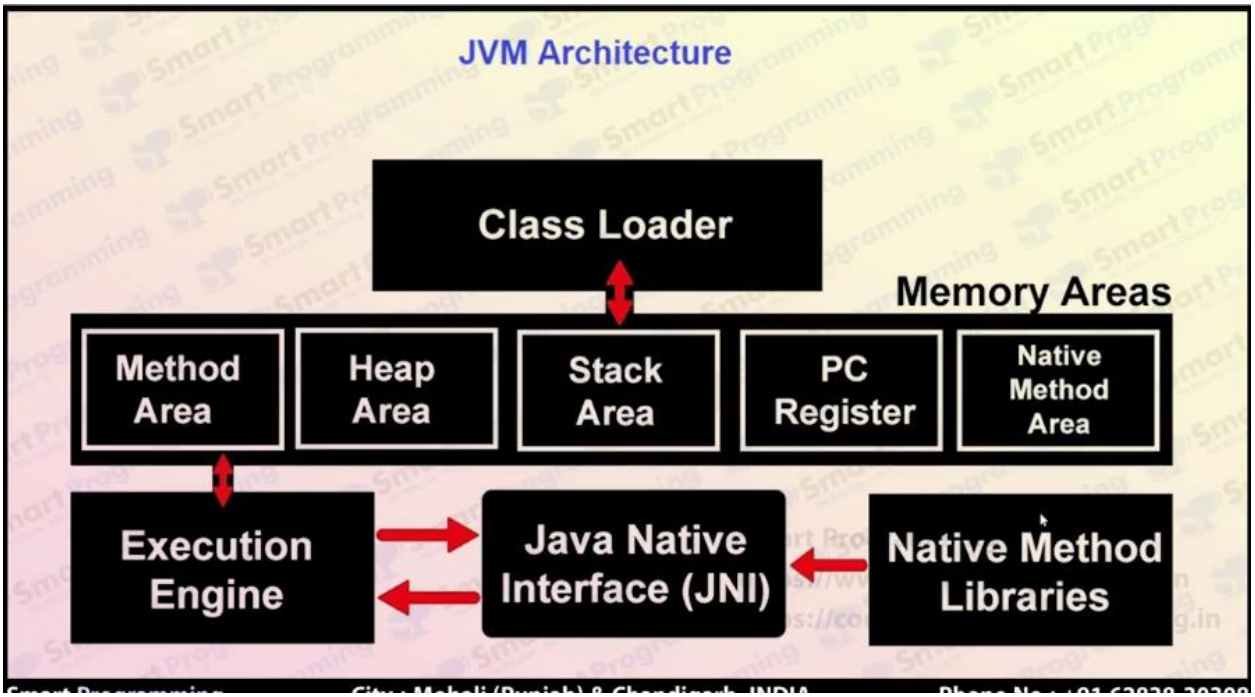
JVM
- class loader
- memory areas
- Execution Engine

Java Package Classes

Runtime Libraries

Development Tools
- java
- javac
- javadocs
- javaw

Smart Programming
https://www.smartprogramming.in
https://courses.smartprogramming.in



JVM Architecture

Class Loader

Memory Areas

Method Area

Heap Area

Stack Area

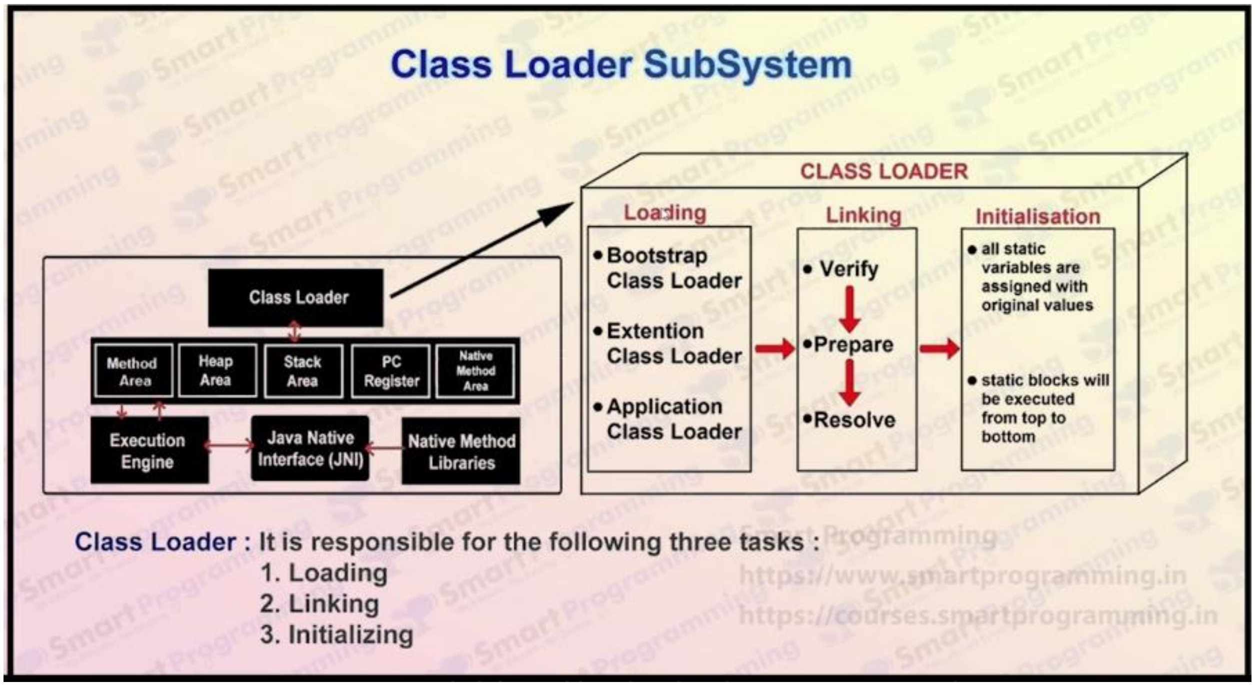
PC Register

Native Method Area

Execution Engine

Java Native Interface (JNI)

Native Method Libraries



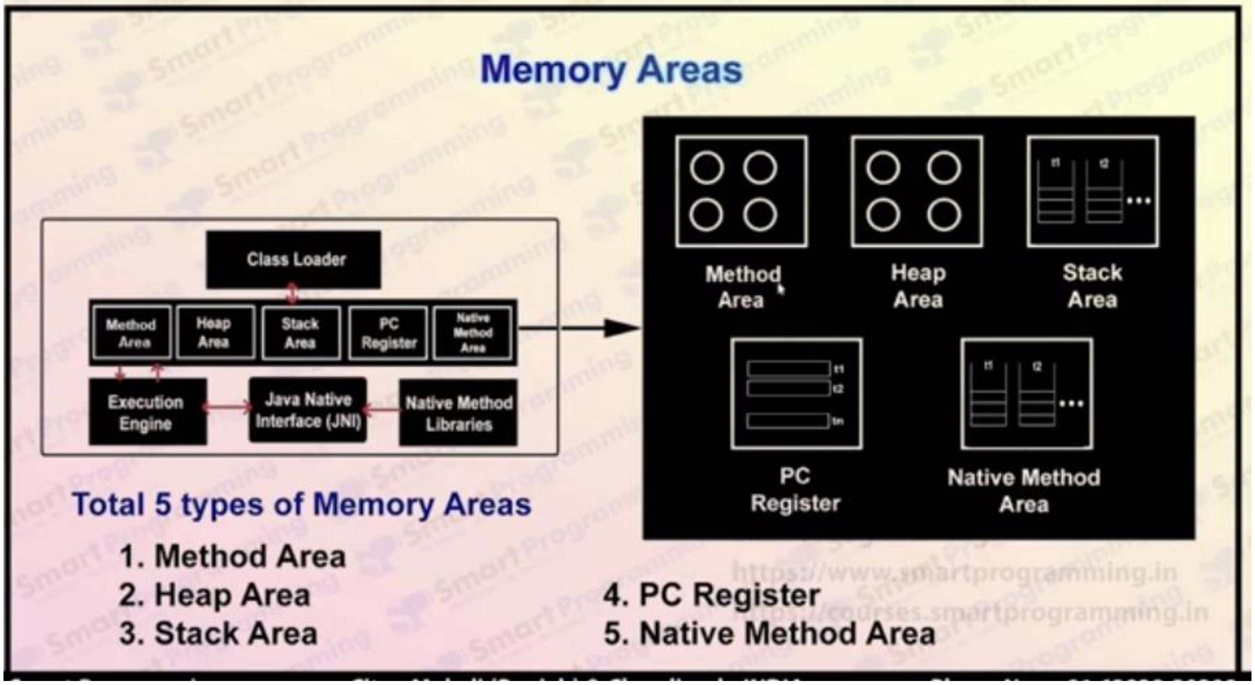
Class Loader SubSystem

Class Loader
Method Area
Heap Area
Stack Area
PC Register
Native Method Area
Execution Engine
Java Native Interface (JNI)
Native Method Libraries
CLASS LOADER
Loading
Linking
Initialisation
• Bootstrap Class Loader
• Extention Class Loader
• Application Class Loader
• Verify
• Prepare
• Resolve
• all static variables are assigned with original values
• static blocks will be executed from top to bottom

Class Loader : It is responsible for the following three tasks :

1. Loading
2. Linking
3. Initializing

https://www.smartprogramming.in
https://courses.smartprogramming.in



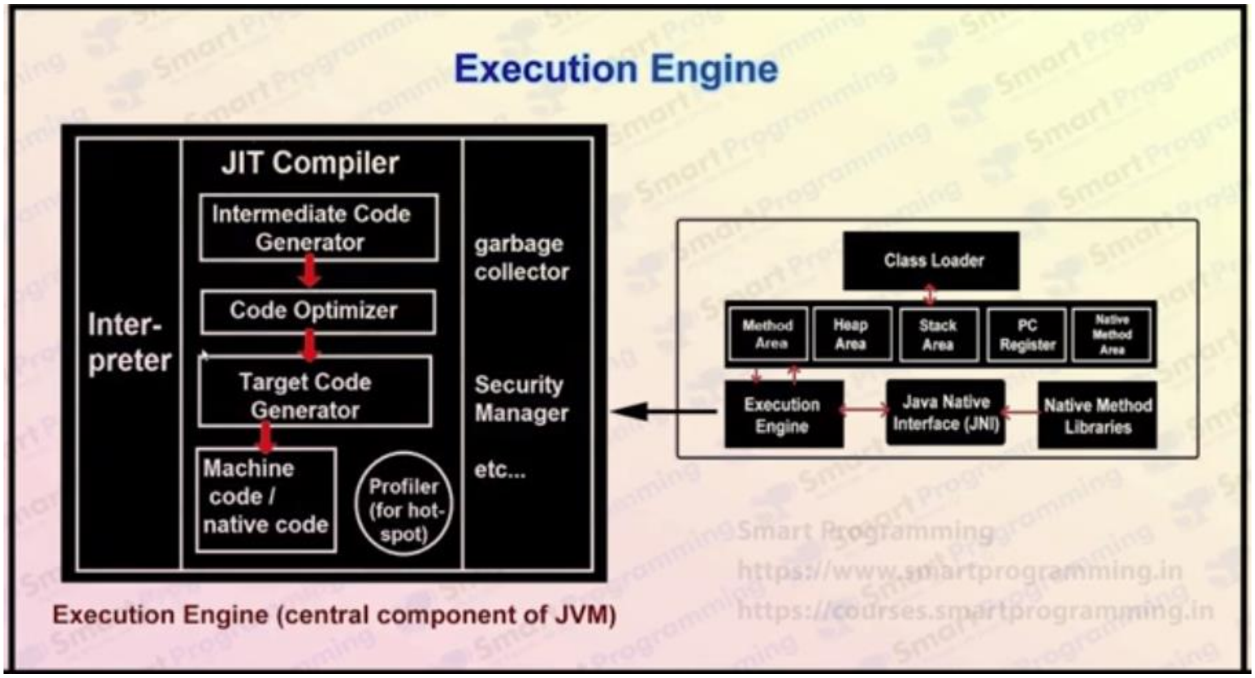
Memory Areas

Class Loader
Method Area
Heap Area
Stack Area
PC Register
Native Method Area
Execution Engine
Java Native Interface (JNI)
Native Method Libraries
t1
t2
tn
Method Area
Heap Area
Stack Area
PC Register
Native Method Area

Total 5 types of Memory Areas

1. Method Area
2. Heap Area
3. Stack Area
4. PC Register
5. Native Method Area

https://www.smartprogramming.in
https://courses.smartprogramming.in



Execution Engine

Inter-
preter

JIT Compiler

Intermediate Code Generator

Code Optimizer

Target Code Generator

Machine code / native code

Profiler (for hot-spot)

garbage collector

Security Manager

etc...

Class Loader

Method Area

Heap Area

Stack Area

PC Register

Native Method Area

Execution Engine

Java Native Interface (JNI)

Native Method Libraries

Execution Engine (central component of JVM)

Smart Programming
https://www.smartprogramming.in
https://courses.smartprogramming.in

Smart Programming : YouTube Channel

An investment in Knowledge pays the best interest….

Smart Programming
We Educate - We Develop

☎ +91 62838-30308
Call us to Learn
Latest Technologies

City : Mohali (Punjab),
& Chandigarh
(India)

WEBSITE : http://www.smartprogramming.in

BUY COURSES ON : https://courses.smartprogramming.in

YOUTUBE CHANNEL : Smart Programming (https://www.youtube.com/c/SmartProgramming)

ANDROID APP : Smart Programming
(https://play.google.com/store/apps/details?id=com.smartprogramming)

https://www.facebook.com/smartprogramming.india

https://www.instagram.com/smart_programming

Java
android
python
Data Science
Machine Learning
Artificial Intelligence
M E A N
Full Stack
BIG DATA
hadoop

Data Types in Java

=> Programming Language Fundamentals / Terminologies :-

- -> Data Types
- -> Variables
- -> Tokens
  - -> Literals
  - -> Operators
  - -> Separators
  - -> Punctuators
  - -> Comments
  - -> Keywords / Reserved Words
  - -> Identifiers

=> Data Types :-

-> The type of data that we are specifing to java is known as Data Type.

-> For example :-

10 - int

'a' - char

"deepak" - String

true - boolean

-> According to data type, languages are divided into 2 categories :-

1. Statically Type Langauges :- In this type of languages we have to specify the type of each data and thus compiler known which type of data we have provided. For examples C, C++, Java, FORTRAIN, Pascal etc

2. Dynamically Typed Langauges :- In this type of languages we dont need to specify the type

of data that we have provided. For examples Python, JavaScript, Objective C, Ruby etc

-> Types of Data Types :-

1. Primitive Data Types (Predefined Data Type) :-

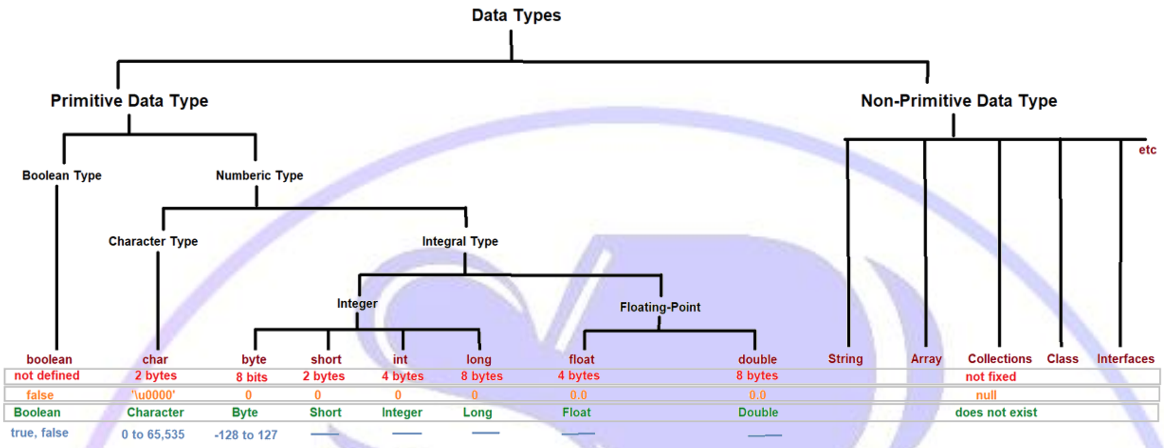
- The data types which are already provided by java and whose size are fixed are know as primitive data type

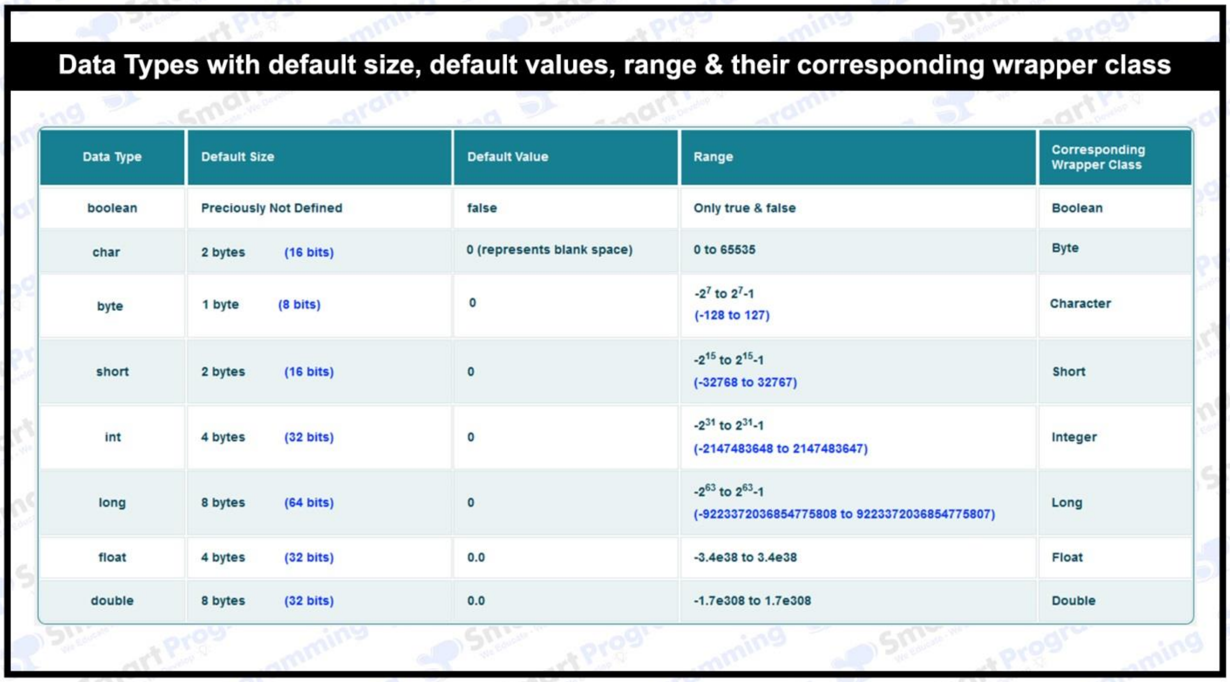
- Examples :- There are 8 primitive data types :-

boolean, char, byte, short, int, long, float, double

- To find the range of Integer primitive data type we can use the formula i.e. -2(n-1) to 2(n-1) - 1 (where n is no of bits)

- To find the range (minimum and maximum value) of primitive data types (excluding boolean) we can use static int variables i.e. MIN_VALUE & MAX_VALUE


Data Types
Primitive Data Type
Non-Primitive Data Type
Boolean Type
Numberic Type
Character Type
Integral Type
Integer
Floating-Point
etc
boolean
char
byte
short
int
long
float
double
String
Array
Collections
Class
Interfaces
not defined
2 bytes
8 bits
2 bytes
4 bytes
8 bytes
4 bytes
8 bytes
not fixed
false
'\u0000'
0
0
0
0
0.0
0.0
null
Boolean
Character
Byte
Short
Integer
Long
Float
Double
does not exist
true, false
0 to 65,535
-128 to 127



Data Types with default size, default values, range & their corresponding wrapper class

Data Type	Default Size	Default Value	Range	Corresponding Wrapper Class
boolean	Preciously Not Defined	false	Only true & false	Boolean
char	2 bytes (16 bits)	0 (represents blank space)	0 to 65535	Byte
byte	1 byte (8 bits)	0	-2^7 to 2^7-1 (-128 to 127)	Character
short	2 bytes (16 bits)	0	-2^{15} to $2^{15}-1$ (-32768 to 32767)	Short
int	4 bytes (32 bits)	0	-2^{31} to $2^{31}-1$ (-2147483648 to 2147483647)	Integer
long	8 bytes (64 bits)	0	-2^{63} to $2^{63}-1$ (-9223372036854775808 to 9223372036854775807)	Long
float	4 bytes (32 bits)	0.0	-3.4e38 to 3.4e38	Float
double	8 bytes (32 bits)	0.0	-1.7e308 to 1.7e308	Double

2. Non-Primitive Data Types (User Defined Data Type or Derived Data Type) :-

- Non-Primitive data types are not pre-defined data types but are created by the programmer. These are sometimes know as "reference variable" or "object reference"

- The size of non-primitive data type is not fixed

- Examples :- String, Array, Collection, Class, Abstract Class, Interface etc

---

Interview Questions :-

1. What are statically and dynamically typed languages ?

2. What is difference between Primitive Data Type & Non-Primitive Data Type

3. What are Wrapper Classes

-> The classes which are used to convert primitive into objects and objects into primitive

-> There are 8 wrapper classes :- Boolean, Character, Byte, Short, Integer, Long, Float & Double

-> Java introduced autoboxing and unboxing in J2SE 5.0 version which coverts primitive into object and object into primtive automatically

4. What is Autoboxing & Unboxing ?

-> Autoboxing is the automatic conversion of primitive data type into its corresponding wrapper classes by java compiler.

-> Unboxing is the automatic conversion of an object of wrapper type to its corresponding primitive value by java compiler

Company Links & Contacts

Company Name: Smart Programming (+91 62838-30308)

Address : Chandigarh & Mohali (Punjab), India

Websites: https://www.smartprogramming.in/
https://courses.smartprogramming.in

Android App:
https://play.google.com/store/apps/details?id=com.smartprogramming

YouTube Channel:
https://www.youtube.com/c/SmartProgramming

Smart Programming : YouTube Channel

An investment in Knowledge pays the best interest….

Smart Programming
We Educate - We Develop

+91 62838-30308
Call us to Learn
Latest Technologies

City : Mohali (Punjab),
& Chandigarh
(India)

WEBSITE : http://www.smartprogramming.in

BUY COURSES ON : https://courses.smartprogramming.in

YOUTUBE CHANNEL : Smart Programming (https://www.youtube.com/c/SmartProgramming)

ANDROID APP : Smart Programming
(https://play.google.com/store/apps/details?id=com.smartprogramming)

https://www.facebook.com/smartprogramming.india

https://www.instagram.com/smart_programming

Java
android
python
Data Science
Machine Learning
Artificial Intelligence
M E A N
Full Stack
BIG DATA
hadoop

Type Casting in Java

=> Data Types More Interview Questions :-

1. What is ADT (Abstract Data Type) :-

-> ADT is a type (or class) which holds the different types of objects with some specifications

-> The defination of ADT only mentions what operations are to be performed but not how these operations implemented

-> For exmaples : Stack ADT, List ADT, Queue ADT etc

2. Why java is not purely OOP's Langauge :-

-> Java is not purely OOP's Langauge because:-

1. Usage of Primitive Data Types
2. Usage of Static members

3. How primitive variables passed to methods - by value or by reference :-

-> Java supports only pass by value

4. What is type casting ?

5. What are different types of Type-Casting in java ?

-> There are 2 types of Type-Casting in Java :-

1. "Primitive Data Type" Type Casting
    1.1 Widening Type Casting (Implicit Type Casting)
    1.2 Narrowing Type Casting (Explicit Type Casting)
2. "User Defined Data Type" Type Casting

6. What is Type Checking & Type Casting ?

-> Type Checking : Type Checking is the responsibility of compiler. It checkes whether the syntax is correct or not and whether we are assigning lower data type to higher data type.

-> Type Casting : Type Casting is the responsibility of JVM. In this phase lower data type (for eg byte) will convert into higher data type (for eg int) and the value will be copied in higher data type

---

=> Type Casting :-

-> The process of converting the data from one data type to another data type is known as type casting

-> There are 2 types of Type-Casting in Java :-

1. "Primitive Data Type" Type Casting

1.1 Widening Type Casting (Implicit Type Casting)

1.2 Narrowing Type Casting (Explicit Type Casting)

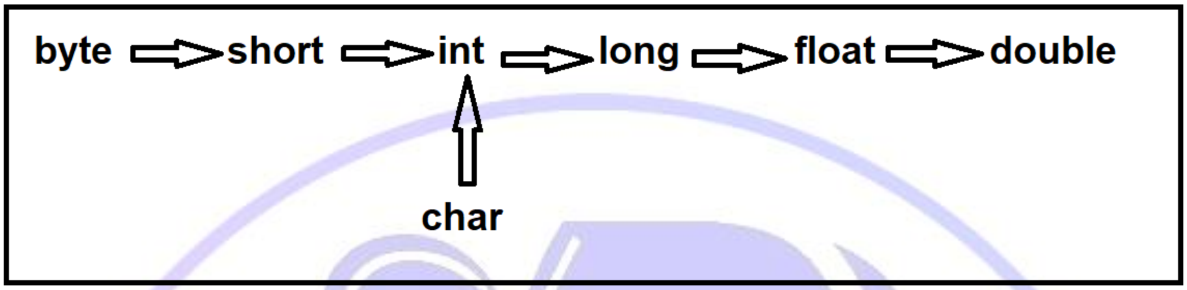
2. "User Defined Data Type" Type Casting

---

1. "Primitive Data Type" Type Casting

1.1 Widening Type Casting (Implicit Type Casting)

-> It is the process of converting data from lower data type to higher data type


byte
short
int
long
float
double
char

Implicit Type Casting (Widening Type Casting)

1.2 Narrowing Type Casting (Explicit Type Casting)

-> In this case we convert higher data type into lower data type

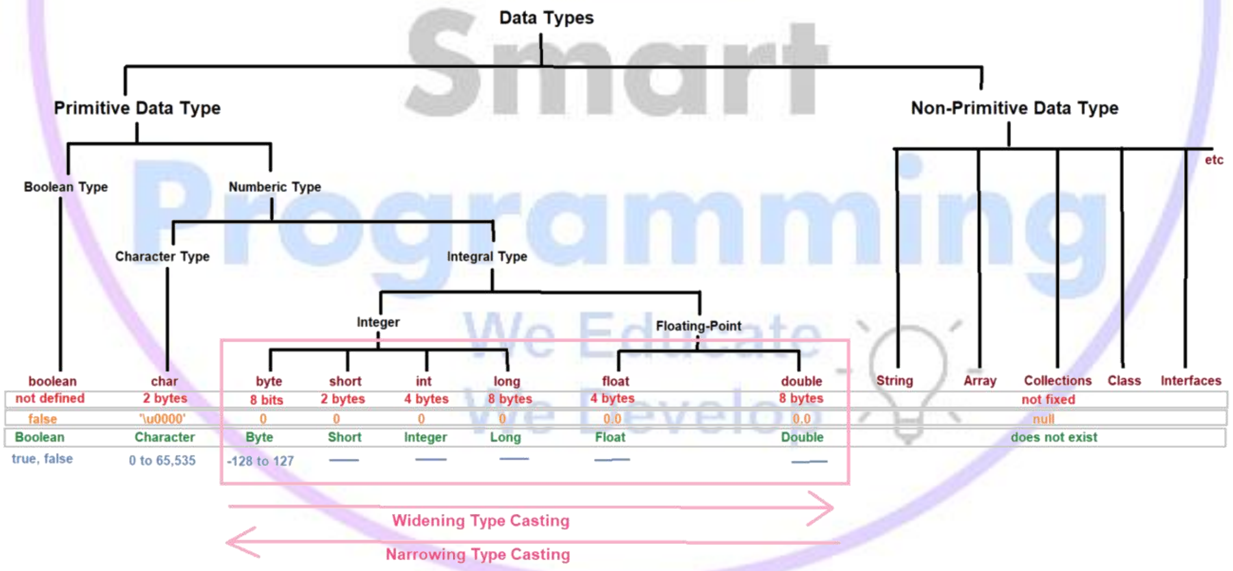
-> Narrowing Type Casting can be achieved by using "cast operator"

=> Note : no1 & no2 , result

no1 + no2 = result

1. If no1 & no2 is byte, short or int then result will be always in int

2. If no1 & no2 is long, float, double etc then result will be in higher data type


Data Types
Primitive Data Type
Non-Primitive Data Type
Boolean Type
Numberic Type
Character Type
Integral Type
Integer
Floating-Point
etc
boolean
not defined
false
Boolean
true, false
char
2 bytes
'\u0000'
Character
0 to 65,535
byte
8 bits
0
Byte
-128 to 127
short
2 bytes
0
Short
int
4 bytes
0
Integer
long
8 bytes
0
Long
float
4 bytes
0.0
Float
double
8 bytes
0.0
Double
String
Array
Collections
Class
Interfaces
not fixed
null
does not exist
Widening Type Casting
Narrowing Type Casting

2. "User Defined Data Type" Type Casting :-

-> It is the process of converting data from one user defined data type to another user defined data type

-> For "user defined data type" type casting, both data types should have relation (either extends or implements)

Company Links & Contacts

Company Name: Smart Programming (+91 62838-30308)

Address : Chandigarh & Mohali (Punjab), India

Websites: https://www.smartprogramming.in/

https://courses.smartprogramming.in

Android App: https://play.google.com/store/apps/details?id=com.smartprogramming

YouTube Channel: https://www.youtube.com/c/SmartProgramming

Smart Programming : YouTube Channel

An investment in Knowledge pays the best interest….

Smart Programming
We Educate - We Develop

☎ +91 62838-30308
Call us to Learn
Latest Technologies

City : Mohali (Punjab),
& Chandigarh
(India)

WEBSITE : http://www.smartprogramming.in

BUY COURSES ON : https://courses.smartprogramming.in

YOUTUBE CHANNEL : Smart Programming (https://www.youtube.com/c/SmartProgramming)

ANDROID APP : Smart Programming
(https://play.google.com/store/apps/details?id=com.smartprogramming)

https://www.facebook.com/smartprogramming.india

https://www.instagram.com/smart_programming

Java
android
python
Data Science
Machine Learning
Artificial Intelligence
M E A N
Full Stack
BIG DATA
hadoop

Number System in Java

=> Number System :-

-> We i.e. Human beings communicate in words, characters and numbers but computers communicate only in numbers.

-> The data or character that we provide to computer is converted into number (In java all characters have unique integer value known as ASCII)

-> Computer (Programming Languages) communicates only in numbers thus there are 4 number systems:-

1. Binary Numer System (Base 2)
2. Octal Number System (Base 8)
3. Decimal Number System (Base 10)
4. Hexa-Decimal Number System (Base 16)

-> Java supports all number systems but default is "Decimal Number System"

1. Binary Numer System (Base 2) :-

-> Alphabet : 0,1

-> Prefix : 0b or 0B

2. Octal Number System (Base 8) :-

-> Alphabet : 0,1,2,3,4,5,6,7

-> Prefix : 0 (zero)

3. Decimal Number System (Base 10) :-

-> Alphabet : 0,1,2,3,4,5,6,7,8,9

-> Prefix : does not have any prefix

4. Hexa-Decimal Number System (Base 16) :-

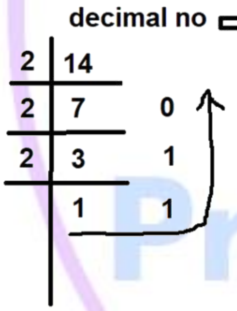
-> Alphabet : 0,1,2,3,4,5,6,7,8,9,A (10), B (11), C (12), D (13), E (14), F (15)

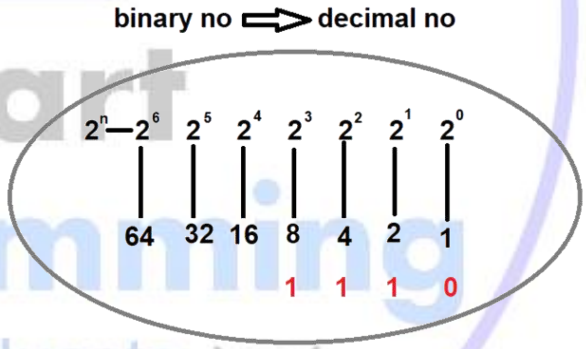
-> Prefix : 0x or 0X

=> Conversion :-

1. If we want to convert Binary or Octal or Hexa-Decimal number system to Decimal number system -> Integer.parseInt("no" , base);

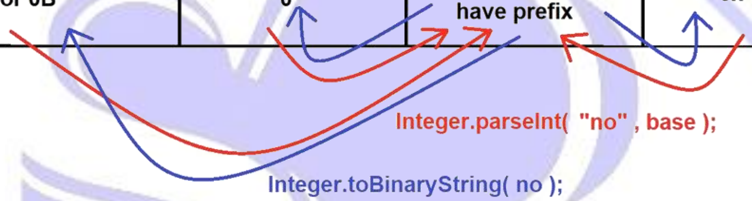
2. If we want to convert Decimal number system to Binary or Octal or Hexa-Decimal number system -> Integer.toXxxString(no);


decimal no ⇒ binary no
1110
2 14
2 7 0
2 3 1
1 1


binary no ⇒ decimal no
2^n—2^6 2^5 2^4 2^3 2^2 2^1 2^0
64 32 16 8 4 2 1
1 1 1 0

Number System & Conversion in Java

	Binary Number System	Octal Number System	Decimal Number System	Hexa-Decimal Number System
Alphabet	0, 1	0,1,2,3,4,5,6,7	0,1,2,3,4,5,6,7,8,9	0,1,2,3,4,5,6,7,8,9, A,B,C,D,E,F
Prefix	0b or 0B	0	does not have prefix	0x or 0X



Integer.parseInt("no" , base);

Integer.toBinaryString(no);
Integer.toOctalString(no);
Integer.toHexString(no);

Company Links & Contacts

Company Name: Smart Programming (+91 62838-30308)

Address : Chandigarh & Mohali (Punjab), India

Websites: https://www.smartprogramming.in/

https://courses.smartprogramming.in

Android App: https://play.google.com/store/apps/details?id=com.smartprogramming

YouTube Channel: https://www.youtube.com/c/SmartProgramming

Smart Programming : YouTube Channel

An investment in Knowledge pays the best interest….

Smart Programming
We Educate - We Develop

☎ +91 62838-30308
Call us to Learn
Latest Technologies

City : Mohali (Punjab),
& Chandigarh
(India)

WEBSITE : http://www.smartprogramming.in

BUY COURSES ON : https://courses.smartprogramming.in

YOUTUBE CHANNEL : Smart Programming (https://www.youtube.com/c/SmartProgramming)

ANDROID APP : Smart Programming
(https://play.google.com/store/apps/details?id=com.smartprogramming)

https://www.facebook.com/smartprogramming.india

https://www.instagram.com/smart_programming

C
C++
C#
Java
android
python
Data Science
Machine Learning
Artificial Intelligence
M E A N
Full Stack
BIG DATA
hadoop

Variables in Java

=> Variable :

-> Variable is the name of memory location that contains the data

-> The variable value can changed according to programming logic

-> Every variable has its data type

-> Examples :

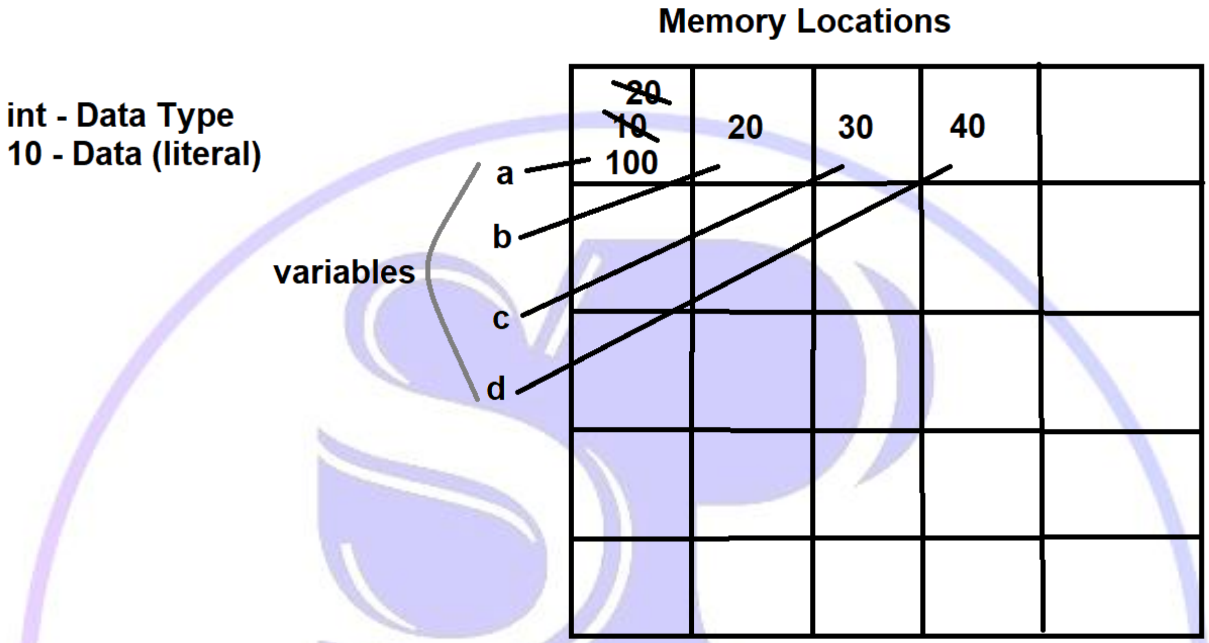
int a=10; (a is the variable)

char c='a'; (c is the variable)

String name="deepak"; (name is the variable)

-> Types of variables :-

1. Local Variables
2. Instance Variables
3. Static Variables


Memory Locations
int - Data Type
10 - Data (literal)
~~20~~
~~10~~
100
20
30
40
a
b
c
d
variables

1. LOCAL VARIABLES :-

-> Declaration : Local variables are declared within the body of methods, constructors or blocks

-> Scope : Local variables can be used within the methods or constructors or blocks but not outside them

-> When local variables gets memory allocated : Local variables gets allocated when the methods or constructors or blocks are executed and get get deleted from memory when that method or block or constructor execution completes

-> Stored Memory Area : Local variables gets memory allocated in "STACK AREA"

-> Default Values : Local variables does not have any default value, if we dont provide the value for local variables and use them, it will provide compile time error saying "variable variable_name might not have been initialized"

-> Access Modifiers : We cannot use access modifiers i.e. public, protected and private with local variables

-> How to access Local Variables :

1. directly

2. INSTANCE VARIABLES :-

-> Declaration : Instance variables are declared within the class but outside the methods or constructors or blocks

-> Scope : Instance variables can be used within the class and every method or block or constructor but not inside the static methods or static blocks

-> When instance variables gets memory allocated : Whenever new object is created, instance variables gets memory allocated and when that object is destroyed instance variables also gets deleted

-> Stored Memory Area : Instance variables are stored in "HEAP AREA"

-> Default Values : Instance variables have default values for ex int - 0; boolean - false; float - 0.0; etc

-> Access Modifiers : We can use access modifiers i.e. public, protected and private with instance variables

-> How to access Instance Variables :

1. directly
2. by using object name

3. STATIC VARIABLES :-

-> Declaration : Static variables are also declared within the class but outside the methods or constructors or blocks and we also use "static" keyword with them

-> Scope : Static variables can be used in every methods or static methods or blocks or static blocks or constructors

-> When static variables gets memory allocated : When we run java program, .class file or byte code gets loaded in JVM and at that time only static variables also gets memory allocated. And when the .class file gets unloaded from JVM static variables gets destroyed from the memory

-> Stored Memory Area : Static variables are stored in "METHOD AREA"

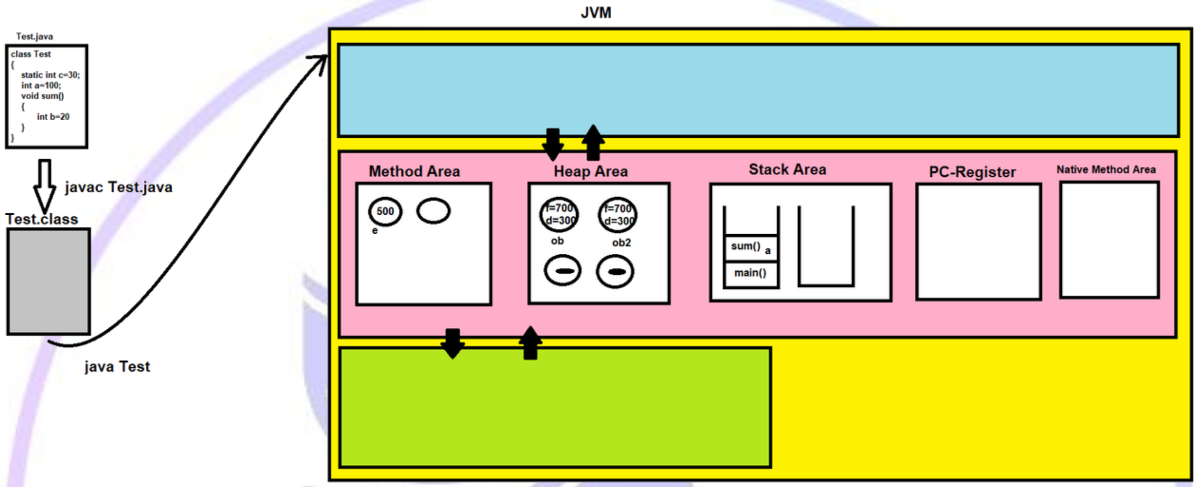
-> Default Values : Static variables have default values for ex int - 0; boolean - false; float - 0.0; etc

-> Access Modifiers : We can use access modifiers i.e. public, protected and private with static variables

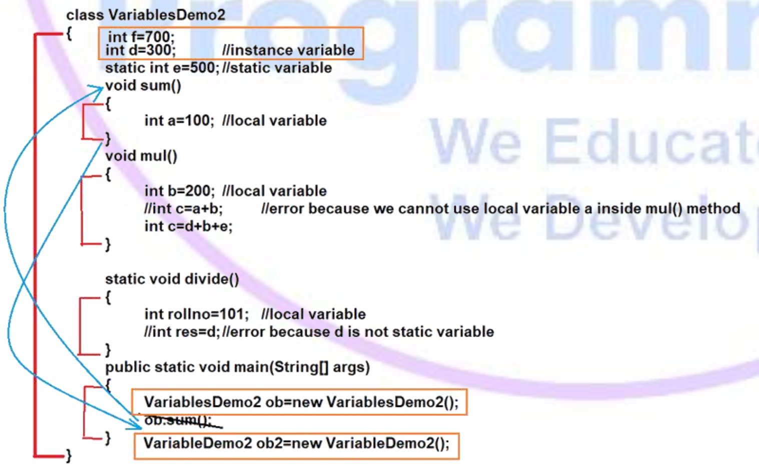
-> How to access Static Variables :

1. directly
2. by object name
3. by class name

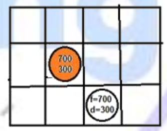
Memory Allocation of local, instance & static variables


JVM
Test.java
class Test
{
static int c=30;
int a=100;
void sum()
{
int b=20
}
}
javac Test.java
Test.class
java Test
Method Area
500
e
Heap Area
f=700
d=300
ob
f=700
d=300
ob2
Stack Area
sum()
a
main()
PC-Register
Native Method Area

Program flow



```
class VariablesDemo2
{
    int f=700;
    int d=300;          //instance variable
    static int e=500; //static variable
    void sum()
    {
        int a=100;  //local variable
    }
    void mul()
    {
        int b=200;  //local variable
        //int c=a+b;        //error because we cannot use local variable a inside mul() method
        int c=d+b+e;
    }

    static void divide()
    {
        int rollno=101;   //local variable
        //int res=d; //error because d is not static variable
    }
    public static void main(String[] args)
    {
        VariablesDemo2 ob=new VariablesDemo2();
        ~~ob.sum();~~
    }
        VariableDemo2 ob2=new VariableDemo2();
}
```


700
300
f=700
d=300

Company Links & Contacts

Company Name: Smart Programming (+91 62838-30308)

Address : Chandigarh & Mohali (Punjab), India

Websites: https://www.smartprogramming.in/

https://courses.smartprogramming.in

Android App: https://play.google.com/store/apps/details?id=com.smartprogramming

YouTube Channel: https://www.youtube.com/c/SmartProgramming

Smart Programming : YouTube Channel

An investment in Knowledge pays the best interest….

Smart Programming
We Educate - We Develop

+91 62838-30308
Call us to Learn
Latest Technologies

City : Mohali (Punjab),
& Chandigarh
(India)

WEBSITE : http://www.smartprogramming.in

BUY COURSES ON : https://courses.smartprogramming.in

YOUTUBE CHANNEL : Smart Programming (https://www.youtube.com/c/SmartProgramming)

ANDROID APP : Smart Programming
(https://play.google.com/store/apps/details?id=com.smartprogramming)

https://www.facebook.com/smartprogramming.india

https://www.instagram.com/smart_programming

Java
android
python
Data Science
Machine Learning
Artificial Intelligence
M E A N
Full Stack
BIG DATA
hadoop

Literals & Operators in Java

=> Tokens :-

-> Tokens are the smallest unit or say small building blocks of java program that are meaningful to the java compiler

-> For example :System.out.println("hello");

Tokens are : System, ., out, println, (,), hello, ;

-> Our java program converts into tokens and then java compiler converts these tokens into java bytecode

-> Different types of tokens are :-

1. Literals
2. Operators
3. Separators
4. Punctuators
5. Comments
6. Keywords/Reserved Words
7. Identifiers

1. Literals :

-> Any contant value assigned to the variable is known as literal

-> For example :-

```
int a=10;     //10 is literal
char c='x';   //x is literal
String name="deepak";  //deepak is literal
boolean b=true;
```

-> Types of literals :-

1. Intergal Literals (byte, short, int, long)
2. Floating-Point Literals (float, double)
3. Character Literals (Single Quotes [''], Char literal in Integer literal, unicode representation, escape character)
4. String Literals
5. Boolean Literals

2. Operators :-

-> Operators are the special symbols which are used to perform any operation on one or more operands

-> For example :-

c=a+b; (+ is operator and a,b are operands)

-> Types of operators :-

1. Arithmetic Operators

+, -, *, /, %

2. Unary Operators

posfix : no++, no--

prefix : ++no, --no, +no, -no, ~no, !no

3. Assignment Operators

=, +=, -=, *=, /=, &=, ^=, |=, <<=, >>=, >>>=

4. Relational Operators

==, !=, <, >, <=, >=, instanceof

(provides the output in true or false)

5. Bitwise Operators

-> Binary Bitwise Operators

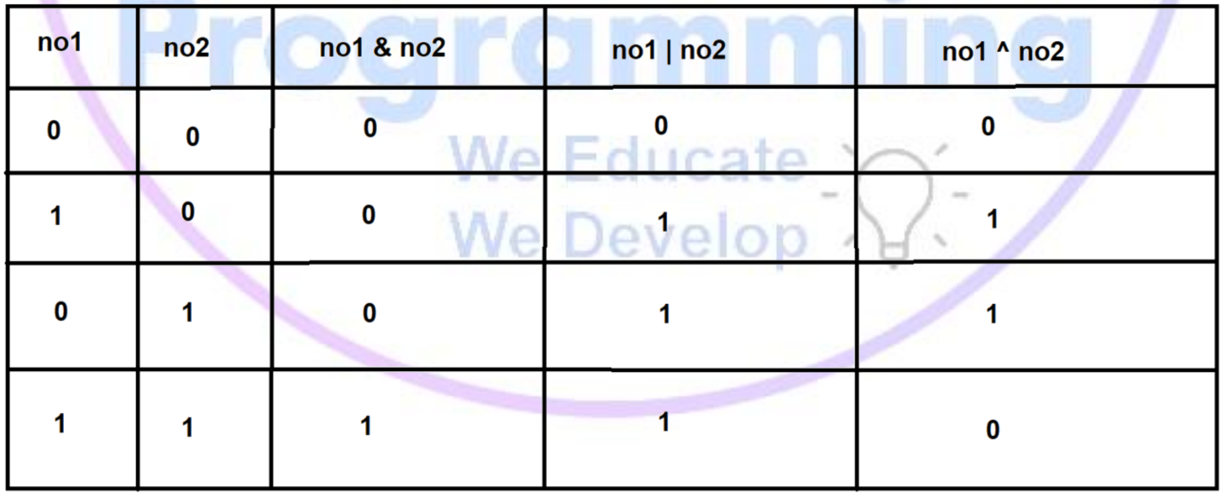
1. Bitwise Logical Operators

& (bitwise AND), |(bitwise OR), ^(bitwise exclusive OR)

2. Shift Operators

>> (right shift), << (left shift), >>> (zero fill right shift)

Bitwise Operator Table



no1	no2	no1 & no2	no1 | no2	no1 ^ no2
0	0	0	0	0
1	0	0	1	1
0	1	0	1	1
1	1	1	1	0

-> Unary Bitwise Operators

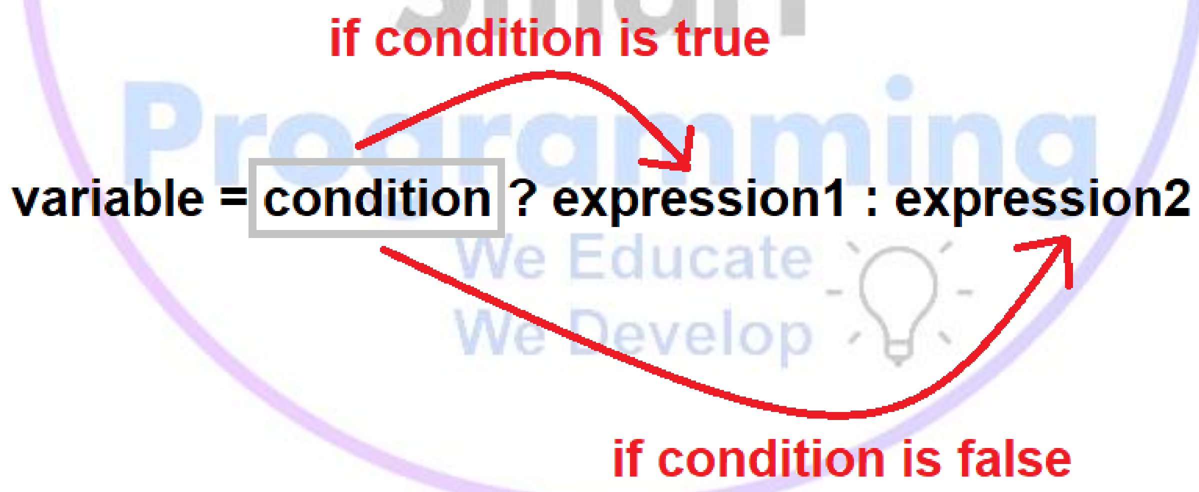
One's compliment Operator

6. Logical Operators

&& (logical AND), || (logical OR)

7. Ternary Operator

?: (variable = condition ? expression 1 : expression2)


if condition is true
variable = condition ? expression1 : expression2
if condition is false

Interview Questions :-

1. What is instanceof operator ?

-> This is used to verify if the specified object is the instance of specified class

2. What is difference between & and && operator?

-> & - Bitwise AND

&& - Logical AND

-> & - Operates on bit values

&& - Operates on boolean values

-> & - In this case both sides will get evaluated because there is no true or false case here

&& - In this case if first expression is false, then it will not evaluate the second condition and it will return false directly

-> & - Used in bit manipulation

&& - Used in conditional statements and loops

3. What is ternary operator ?

Company Links & Contacts

Company Name: Smart Programming (+91 62838-30308)

Address : Chandigarh & Mohali (Punjab), India

Websites: https://www.smartprogramming.in/

https://courses.smartprogramming.in

Android App: https://play.google.com/store/apps/details?id=com.smartprogramming

YouTube Channel: https://www.youtube.com/c/SmartProgramming

Smart Programming : YouTube Channel

An investment in Knowledge pays the best interest….

Smart Programming

We Educate - We Develop

+91 62838-30308

Call us to Learn Latest Technologies

City : Mohali (Punjab), & Chandigarh (India)

WEBSITE : http://www.smartprogramming.in

BUY COURSES ON : https://courses.smartprogramming.in

YOUTUBE CHANNEL : Smart Programming (https://www.youtube.com/c/SmartProgramming)

ANDROID APP : Smart Programming (https://play.google.com/store/apps/details?id=com.smartprogramming)

https://www.facebook.com/smartprogramming.india

https://www.instagram.com/smart_programming

Java
android
python
Data Science
Machine Learning
Artificial Intelligence
M E A N
Full Stack
BIG DATA
hadoop

Keywords & Identifiers in Java

=> Separators :-

-> A separator is a symbol that is used to separate a group of code from one another.

-> Separator help to define the java program structure

-> Types of separators :-

1. () - Parentheses :

   > Encloses the arguments in method definations and calling

   > Adujst the precedence in arithmetic expressions

   > Used in testing the expressions in control statements

   > Used in casting

2. {} - Braces :

   > Used to define the block of class or methods or blocks

   > Used to put values in arrays

3. [] - Brackets :

> Used to declare an arrays

> Used when dereferencing array values

4. ; - Semicolon :

> Used to separate or terminate the statements

5. , - Comma :

> Used to separate the variables declaration

> Used to chain statements together inside a for loop

6. . - Period

> Used to select methods or fields from an object

> Used to separate package name with sub-package name and class name

=> Punctuators :-

-> Punctuators are the symbols or tokens that has semantic meaning to the compiler

-> Types of punctuators :-

1. ? - Question Mark

> Used in ternary operator

2. : - Colon

> Used after loop labels

3. :: - Double Colon

> Used to create method or constructor references

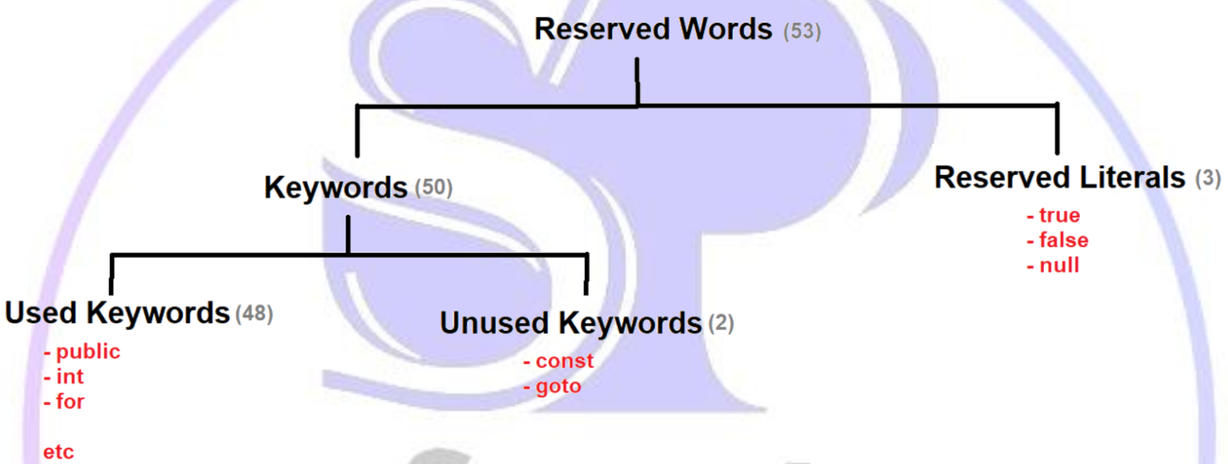
=> Comments :-

-> There are three types of comments :-

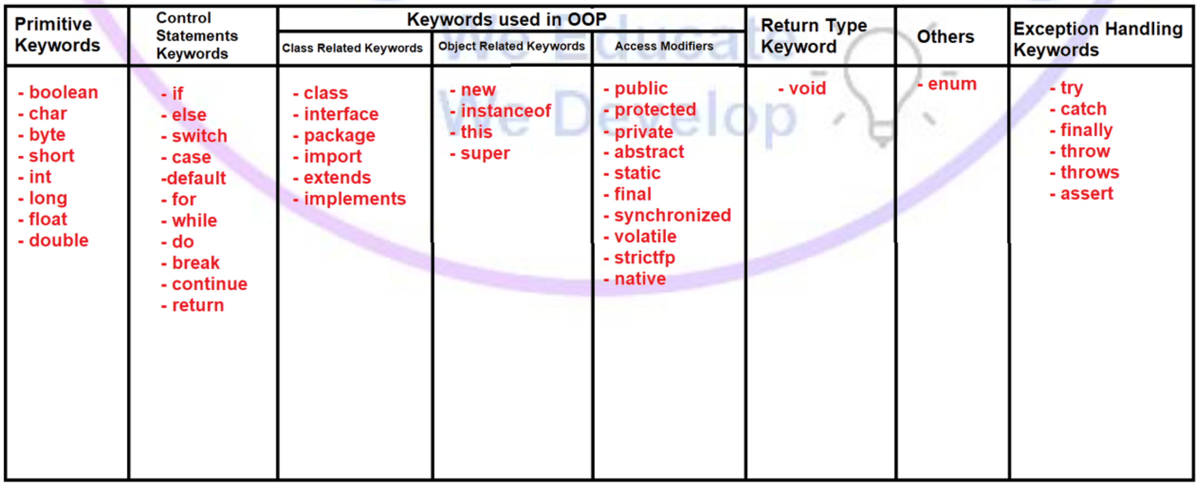
1. Single Line Comment - //
2. Multiline Comment - /* ------ */
3. Documentation Comment - /* *- *- *- *- */

=> Keywords/Reserved Words :-

-> Keywords are the predefined words having any specific meaning


Reserved Words (53)
Keywords (50)
Reserved Literals (3)
- true
- false
- null
Used Keywords (48)
- public
- int
- for
etc
Unused Keywords (2)
- const
- goto

List of Keywords in Java



Primitive Keywords	Control Statements Keywords	Keywords used in OOP			Return Type Keyword	Others	Exception Handling Keywords
		Class Related Keywords	Object Related Keywords	Access Modifiers			
- boolean - char - byte - short - int - long - float - double	- if - else - switch - case -default - for - while - do - break - continue - return	- class - interface - package - import - extends - implements	- new - instanceof - this - super	- public - protected - private - abstract - static - final - synchronized - volatile - strictfp - native	- void	- enum	- try - catch - finally - throw - throws - assert

=> Identifiers :-

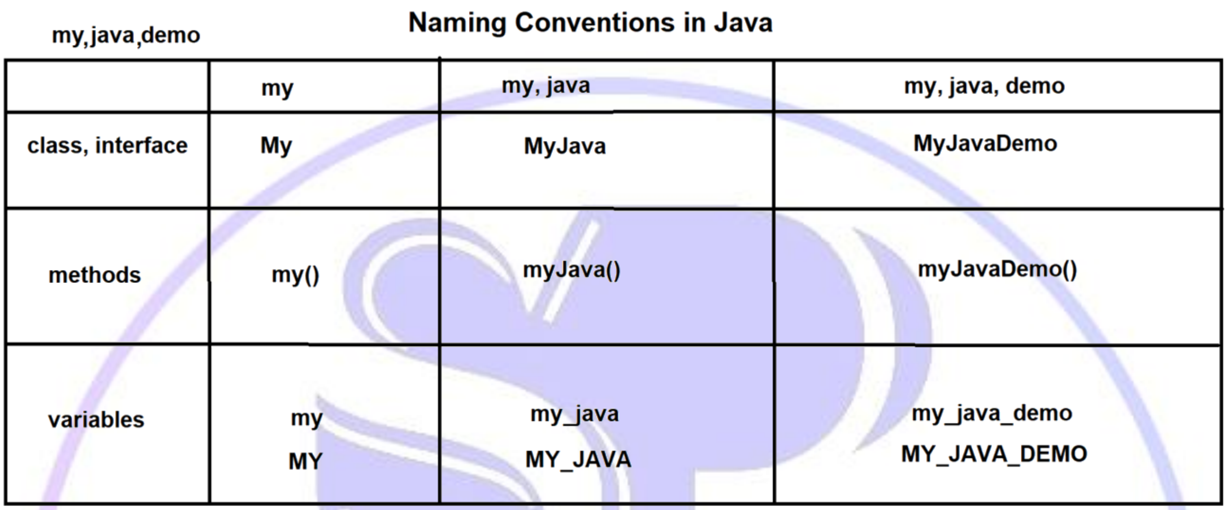
-> Identifier is any name, it can be variable name or class name or interface name or method name or package name etc.

-> Rules for identifiers :-

1. Spaces cannot be used
2. Only two symbols i.e. _ and $ can be used in identifiers name
3. Integer values cannot be used at first position but can be used after first character
4. Reserved words cannot be used as an identifier name

-> Naming Conventions :-

Explained using image in next page



my,java,demo

Naming Conventions in Java

	my	my, java	my, java, demo
class, interface	My	MyJava	MyJavaDemo
methods	my()	myJava()	myJavaDemo()
variables	my MY	my_java MY_JAVA	my_java_demo MY_JAVA_DEMO

1. Which is correct
    1. import or imports
    2. implement or implements
    3. extend or extends
    4. instanceof or instanceOf

Company Links & Contacts

Company Name: Smart Programming (+91 62838-30308)

Address : Chandigarh & Mohali (Punjab), India

Websites: https://www.smartprogramming.in/

https://courses.smartprogramming.in

Android App:
https://play.google.com/store/apps/details?id=com.smartprogramming

YouTube Channel:
https://www.youtube.com/c/SmartProgramming

Smart Programming
We Educate - We Develop

+91 62838-30308
Call us to Learn
Latest Technologies

City : Mohali (Punjab),
& Chandigarh
(India)

WEBSITE : http://www.smartprogramming.in

BUY COURSES ON : https://courses.smartprogramming.in

YOUTUBE CHANNEL : Smart Programming (https://www.youtube.com/c/SmartProgramming)

ANDROID APP : Smart Programming
(https://play.google.com/store/apps/details?id=com.smartprogramming)

https://www.facebook.com/smartprogramming.india

https://www.instagram.com/smart_programming

Java
android
python
Data Science
Machine Learning
Artificial Intelligence
M E A N
Full Stack
BIG DATA
hadoop

Selection Statements in Java

=> Control Statements :-

-> Control Statements are those who can control the flow of the program

-> Types of Control Statements :-

1. Selection Statements

    if, if else, if-else if, nested if, switch

2. Iteration/Looping Statements

    for, while, do-while, for-each loop

3. Jump Statements

    break, continue, return

=> There are 2 types of conditional expressions :-

1. Constant Expression

-> Constant expression are those in which we provide only constant values or final variables

-> Constant expressions are evaluated by "Compiler"

2. Variable Expression

-> Variable expression are those in which we provide the variables

-> Variable expressions are evaluated by "JVM"

=> Selection Statements :-

1. if :-

```
Syntax : if(condition)      //condition -> boolean
         {
               statements;
         }
```

2. if else :-

```
Syntax : if(condition)
         {
               statements;
         }
         else
         {
               statements
}
```

3. If else-if (ladder if):-

```
Syntax : if(condition)
         {
              statements;
         }
         else if(condition)
         {
              statements;
         }
         else if(condition)
         {
              statements;
         }
         -
```

```
-
-
else
{
      statements;
}
```

4. Nested if :-

```
Syntax :- if(condition)
           {
                   if(condition)
                   {
                          if(condition)
                          {


                          }
```

```
                }
        }
        else
        {
                if(condition)
                {

                }
        }
```

5. switch :-

```
Syntax : switch(variable to be tested)
         {
              case value1 : Statements;
                            break;
              case value2 : Statements;
```

```
                              break;
                    case value3 : Statements
                              break;
                    -
                    -
                    -
                    default : Statements;
                           break;
                }
```

NOTE : variable to be tested must be byte, short, int, char and String but not long, float, double, boolean

Programs :-

1. WAP to find the greater number between 2 numbers

2. WAP to find the greater number between 3 numbers

3. WAP to check for leap year

4. WAP to create simple calculator using switch statement

5. WAP to find the greater number between 2 numbers using ternary operator

6. WAP to swap the 2 numbers

7. WAP to swap the 2 numbers without using third variable

Company Links & Contacts

Company Name: Smart Programming (+91 62838-30308)

Address : Chandigarh & Mohali (Punjab), India

Websites: https://www.smartprogramming.in/

https://courses.smartprogramming.in

Android App: https://play.google.com/store/apps/details?id=com.smartprogramming

YouTube Channel: https://www.youtube.com/c/SmartProgramming

Smart Programming : YouTube Channel

An investment in Knowledge pays the best interest….

Smart Programming
We Educate - We Develop

+91 62838-30308
Call us to Learn
Latest Technologies

City : Mohali (Punjab),
& Chandigarh
(India)

WEBSITE : http://www.smartprogramming.in

BUY COURSES ON : https://courses.smartprogramming.in

YOUTUBE CHANNEL : Smart Programming (https://www.youtube.com/c/SmartProgramming)

ANDROID APP : Smart Programming
(https://play.google.com/store/apps/details?id=com.smartprogramming)

https://www.facebook.com/smartprogramming.india

https://www.instagram.com/smart_programming

Java
ANDROID
python
Data Science
Machine Learning
Artificial Intelligence
M E A N
Full Stack
BIG DATA
hadoop

Iteration/Looping Statements in Java

=> Control Statements :-

-> Control Statements are those who can control the flow of the program

-> Types of Control Statements :-

1. Selection Statements

   if, if else, if-else if, nested if, switch

2. Iteration/Looping Statements

   for, while, do-while, for-each loop

3. Jump Statements

   break, continue, return

=> More concepts for if part

-> Case 1 : If there is only single statement in if or else then its not compulsory to use curly braces {}

-> Case 2 : If there is no statement in if or else part then we have to use either curly braces - {} or semi-colon - ;

-> Case 3 : The syntax of "if" is "if(expression){}", then expression should return boolean value. In expression we can provide direct condition or we can also provide method calling statement which returns boolean value

=> Iteration/Looping Statements :-

=> for :

Syntax : for(expression1; expression2; expression3)

```
{
    statements;
}
```

where:

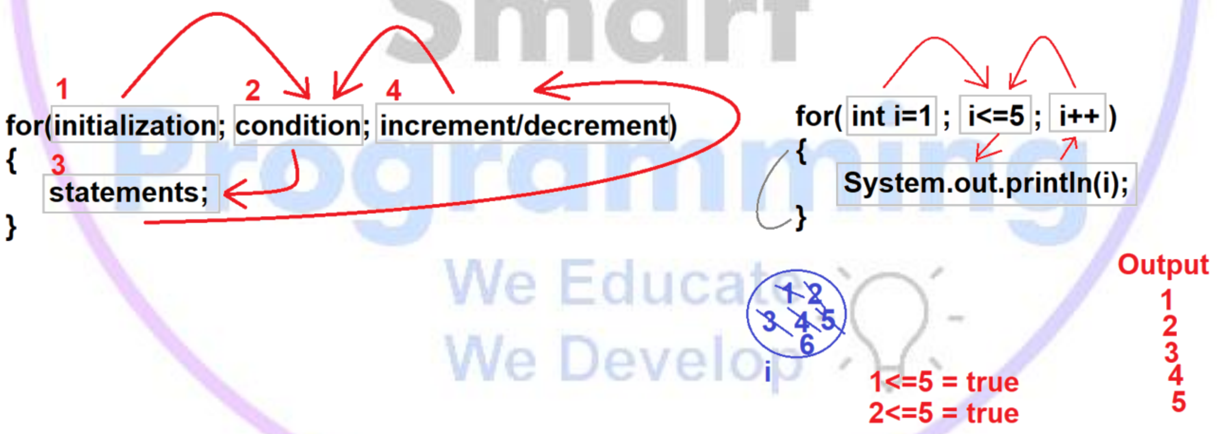
-> expression1 should be any java valid statement but prefered is declaration and initialization

-> expression2 should always be conditional statement which should return boolean value

-> expression3 should be any java valid statement but prefered is increment or decrement simple for of for loop :

```
for(initialization; condition; increment/decrement)
{
        statements;
}
```

Working :



```
for(initialization; condition; increment/decrement)
{
  statements;
}
```

```
for( int i=1 ; i<=5 ; i++ )
{
  System.out.println(i);
}
```

1<=5 = true
2<=5 = true

Output
1
2
3
4
5

Cases :

-> Case 1 : If we dont declare the expression1 then also program will compile fine

-> Case 2 : If we dont declare expression2 then boolean statement will be always true.

-> Case 3 : If we dont declare expression3 then there will not be any increment or decrement

---

Programs :-

1. WAP to print any table
2. What are unrechable statements ?

Company Links & Contacts

Company Name: Smart Programming (+91 62838-30308)

Address : Chandigarh & Mohali (Punjab), India

Websites: https://www.smartprogramming.in/

https://courses.smartprogramming.in

Android App:
https://play.google.com/store/apps/details?id=com.smartprogramming

YouTube Channel:
https://www.youtube.com/c/SmartProgramming

Smart Programming : YouTube Channel

An investment in Knowledge pays the best interest….

Smart Programming

We Educate - We Develop

☎ +91 62838-30308

Call us to Learn
Latest Technologies

City : Mohali (Punjab),
& Chandigarh
(India)

WEBSITE : http://www.smartprogramming.in

BUY COURSES ON : https://courses.smartprogramming.in

YOUTUBE CHANNEL : Smart Programming (https://www.youtube.com/c/SmartProgramming)

ANDROID APP : Smart Programming
(https://play.google.com/store/apps/details?id=com.smartprogramming)

https://www.facebook.com/smartprogramming.india

https://www.instagram.com/smart_programming

Java
android
python
Data Science
Machine Learning
Artificial Intelligence
M E A N
Full Stack
BIG DATA
hadoop

Iteration/Looping Statements in Java

=> Control Statements :-

-> Control Statements are those who can control the flow of the program

-> Types of Control Statements :-

1. Selection Statements

   if, if else, if-else if, nested if, switch

2. Iteration/Looping Statements

   for, while, do-while, for-each loop

3. Jump Statements

   break, continue, return

=> Iteration/Looping Statements :-

-> while :-

```
Syntax : while(condition)
         {
              statements;
         }
```

When we should use while loop : When we dont know how many time we have to execute the loop then we have to use while loop

Cases :

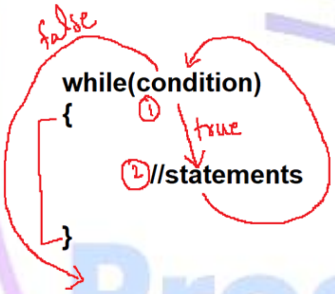
1. Its compulsory to provide the condition in while loop

2. If while loop is infinite and we provide any expression after while loop then it will provide compile time error saying "Unreachable statement"

-> do-while :-

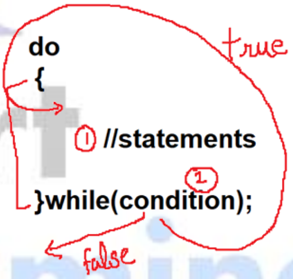
Syntax : do

```
{
    //statements
}while(condition);
```



```
while(condition)
{
        //statements
}
```

false
①
true
②



```
do
{
   //statements
}while(condition);
```

true
①
②
false

Programs :

1. WAP to get the number of digits in integer value (163 - 3; 5673 - 4)

2. WAP to reverse a number (163 -> 361; 5673 -> 3765)

3. WAP to create calculator using do-while loop

=> Interview Question :-

1. What is difference between while and do-while loop

2. How to take user input in java

Company Links & Contacts

Company Name: Smart Programming (+91 62838-30308)

Address : Chandigarh & Mohali (Punjab), India

Websites: https://www.smartprogramming.in/

https://courses.smartprogramming.in

Android App:
https://play.google.com/store/apps/details?id=com.smartprogramming

YouTube Channel:
https://www.youtube.com/c/SmartProgramming

Smart Programming : YouTube Channel

An investment in Knowledge pays the best interest….

Smart Programming

We Educate - We Develop

☎ +91 62838-30308

Call us to Learn Latest Technologies

City : Mohali (Punjab), & Chandigarh (India)

WEBSITE : http://www.smartprogramming.in

BUY COURSES ON : https://courses.smartprogramming.in

YOUTUBE CHANNEL : Smart Programming (https://www.youtube.com/c/SmartProgramming)

ANDROID APP : Smart Programming (https://play.google.com/store/apps/details?id=com.smartprogramming)

https://www.facebook.com/smartprogramming.india

https://www.instagram.com/smart_programming

Java
android
python
Data Science
Machine Learning
Artificial Intelligence
M E A N
Full Stack
BIG DATA
hadoop

Programming Paradigm in Java

=> Programming Paradigm :-

-> Programming paradigm is a way or an approch to solve any problem or to achieve any task using any programming languages

-> Programming paradigm are a way to classify programming languages based on their features

-> There are 2 classifications of programming paradigm :-

1. Imperative Programming Paradigm
2. Declarative Programming Paradigm

=> What is difference between Imperative & Declarative Programming Paradigm ?

1. PROGRAMMING STYLE :-

Imperative :- We have to specify step by step every task

Declarative :- We have to define the problem to achieve the task

2. TASK :-

Imperative :- User makes the decision and command to the compiler

Declarative :- Allows compiler to make decisions

3. REAL WORLD EXAMPLE :-

-> Task (read mails; buy PC; proposal; ....)

Imperative :-

Declarative :-

4. SMART :-

Imperative :- User is more smart as compared to system

Declarative :- System is more smart as compared to user

5. PROGRAMMING FOCUS :-

Imperative :- "What"

Declarative :- "How"

6. PRIMARY FLOW CONTROL :-

Imperative :- Loops, Conditional; functions/methods etc

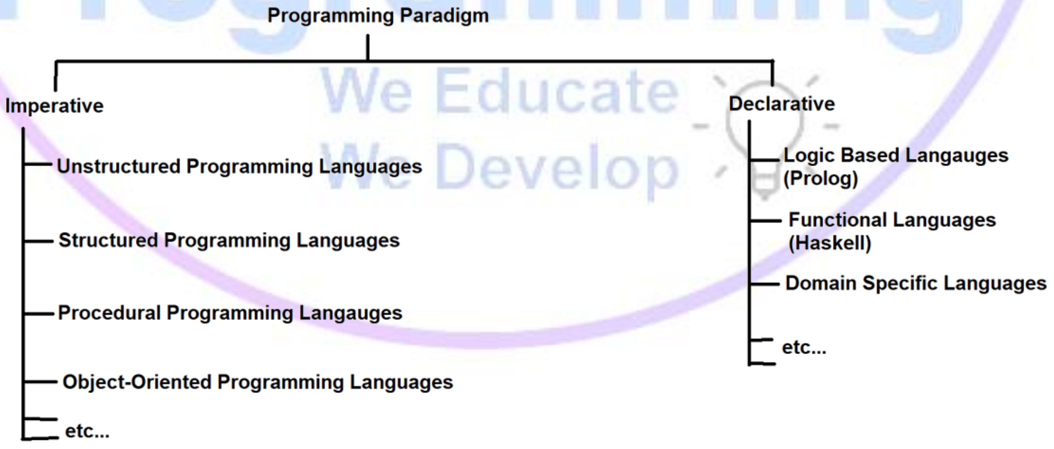
Declarative :- Functions calls (including recursion)

7. EXAMPLE :-

Imperative :- FORTRAN, Assembly Languages, COBOL, C, C++, Java, Python etc

Declarative :- SQL, Haskell, Prolog etc

=> Diagram for Programming Paradigm :-


Programming Paradigm
Imperative
Unstructured Programming Languages
Structured Programming Languages
Procedural Programming Langauges
Object-Oriented Programming Languages
etc...
Declarative
Logic Based Langauges (Prolog)
Functional Languages (Haskell)
Domain Specific Languages
etc...

=> Unstructured Programming Paradigm :-

1. These are the first programming language categories that was introduced at starting point of computers

2. For examples FORTRAN, COBOL, BASIC etc

3. In these languages there was not fixed structure or way to achieve the task or to solve the problem

4. In this type we use mnemonic codes

5. In this part flow control was achieved by "goto" statement

6. Hard to learn and difficult to achieve any task because number of lines of code were increased

=> Structured Programming Paradigm :-

1. These were introduced after unstructured programming paradigm and used till now

2. For examples PASCAL, C etc

3. These have fixed structure to achieve any task

4. These dont use any mnemonic codes which makes this language easier (high level language syntax is used)

5. In this part a lot of flow control statements were introduced for example if, else, for, while etc

6. Easier to learn and easy to achieve any task as there were improved code

-> Structured programming paradigm have a lot of issues like less modularity, abstraction was not good, less security, less sharability, less code reusability and because of these reasons OOP's was introduced

=> Procedure Programming Paradigm :-

1. In POP task can be divided into small parts known as functions/methods

2. In POP top to down approch is used

3. No Access Specifiers

4. It deals with algorithms

5. It uses less memory

6. For example FORTRAN, PASCAL, C etc

-> In POP there were a lot of issues less security and due to this reason OOP's were mostly adopted language by developers

=> Object Oriented Programming Paradigm :-

1. In OOP program is divided into parts i.e. Objects
2. In OOP bottom up approch is used
3. Have a lot of access modifiers or access specifiers
4. OOP deals with data
5. It needs more memory as compared to POP
6. For examples Java, C++, C#, Python etc

NOTE : One language can use multiple programming paradigm

=> What is difference between Object Oriented Programming Languages and Object Based Programming Languages ?

-> In object oriented programming languages inheritance feature is mostly used but in case of object based programming languages inheritance concept is not used.

-> Object oriented programming languages are Java

Object based programming languages are JavaScript

Company Links & Contacts

Company Name: Smart Programming (+91 62838-30308)

Address : Chandigarh & Mohali (Punjab), India

Websites: https://www.smartprogramming.in/

https://courses.smartprogramming.in

Android App:
https://play.google.com/store/apps/details?id=com.smartprogramming

YouTube Channel:
https://www.youtube.com/c/SmartProgramming

Smart Programming : YouTube Channel

An investment in Knowledge pays the best interest….

Smart Programming
We Educate - We Develop

☎ +91 62838-30308
Call us to Learn
Latest Technologies

City : Mohali (Punjab), & Chandigarh (India)

WEBSITE : http://www.smartprogramming.in

BUY COURSES ON : https://courses.smartprogramming.in

YOUTUBE CHANNEL : Smart Programming (https://www.youtube.com/c/SmartProgramming)

ANDROID APP : Smart Programming
(https://play.google.com/store/apps/details?id=com.smartprogramming)

https://www.facebook.com/smartprogramming.india

https://www.instagram.com/smart_programming
Java
android
python
Data Science
Machine Learning
Artificial Intelligence
M E A N
Full Stack
BIG DATA
hadoop

OOP's Concept in Java

(Class, Methods & Objects)

OOP's :-

-> Full form is Object Oriented Programming

-> OOP is the programming paradigm based on the concept of Objects which contains the data(fields or variables) and methods

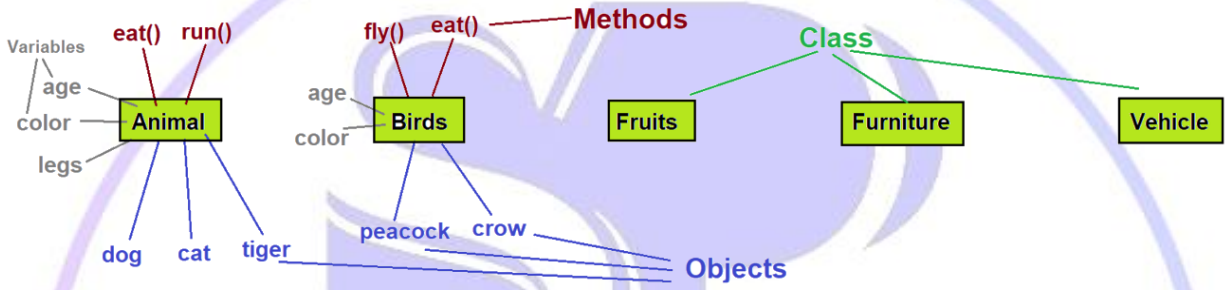
-> It is the most popular programming paradigm used by the programmers

-> For examples : Java, Python, C++ etc

-> Features of OOP :-

1. Class, Objects & Methods
2. Meassage Passing
3. Inheritance & Composition
4. Polymorphism
5. Encapsulation
6. Abstraction

⇨ Real World Example of Class, Methods & Objects


Variables
eat()
run()
age
color
legs
Animal
dog
cat
tiger
fly()
eat()
Methods
age
color
Birds
peacock
crow
Class
Fruits
Furniture
Vehicle
Objects

=> Class :-

-> A class is a user defined blueprint or prototype which is used to create an object

-> Class is a logical entity or say its not a real world entity or class is not physical

-> Real world example :- Animal, Birds, Vehicle, Fruits etc

-> Class represents the set of properties or methods that are common to all the objects of one type

-> Simply we can say that a class is a group of objects having common properties (attributes or variables), behaviour (methods), relationships & semantics.

-> Syntax :

```
access-modifiers class ClassName extends ParentClassName implements InterfaceName
{
    //variables
    //blocks
    //constructors
    //methods
    //nested class, interfaces
}
```

-> Simple syntax :

```
access-modifiers class ClassName
{
	//variables
	//methods

	}
```

-> Simple class

```
class Animal
{
	int age=10;
	String color=black;

}
```

=> Methods :-

-> A set of codes which perform a particular task

-> Advantages :-

1. Code reusability
2. Code optimization

-> Syntax :

```
        access-modifiers return-type
methodName(list of parameters) throws
ExceptionClassName, -, -
        {
                //statements
        }
```

-> Simple Syntax :-

```
        return-type methodName(list of parameters)
        {
                //statements


        }
```

-> Example :-

```
        void eat()
//method declaration
        {
//method defination (body)
                System.out.println("im eating");
        }
```

=> Objects :-

-> Object is an instance of class

-> Object is physical entity or object is real world entity

-> An object has 3 characteristics :-

1. State (represents the data(value) of an object)
2. Behaviour (represents the functionality of an object)
3. Identity (represents the unique id of an object which is created automatically by JVM)

-> Object is simple a memory block

-> Syntax :

1. Creation of an object

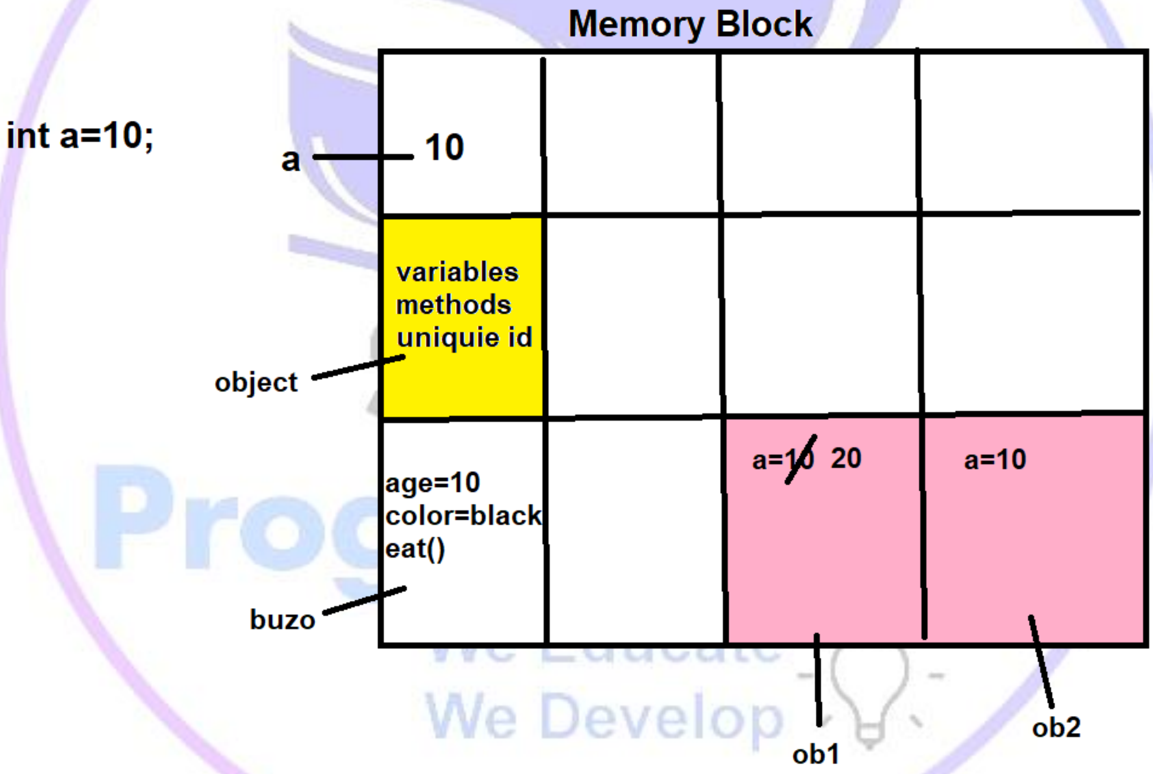
ClassName object_name (ref_variable_name) = new ClassName();

-> Animal buzo = new Animal();

2. Calling variables or methods from object

object_name.variable_name; -> buzo.age;

object_name.methodName(); -> buzo.eat();


Memory Block
int a=10;
a
10
variables
methods
uniquie id
object
age=10
color=black
eat()
buzo
a=~~10~~ 20
a=10
ob1
ob2

=> Points to remember :-

-> We can only use public or default access-modifiers but not private or protected with outer class.

-> For inner class we can use all access-modifiers i.e. public, proctedted, default and private

---

Company Links & Contacts

Company Name: Smart Programming (+91 62838-30308)

Address : Chandigarh & Mohali (Punjab), India

Websites: https://www.smartprogramming.in/

https://courses.smartprogramming.in

Android App: https://play.google.com/store/apps/details?id=com.smartprogramming

YouTube Channel: https://www.youtube.com/c/SmartProgramming

Smart Programming : YouTube Channel

An investment in Knowledge pays the best interest….

Smart Programming
We Educate - We Develop

+91 62838-30308
Call us to Learn
Latest Technologies

City : Mohali (Punjab),
& Chandigarh
(India)

WEBSITE : http://www.smartprogramming.in

BUY COURSES ON : https://courses.smartprogramming.in

YOUTUBE CHANNEL : Smart Programming (https://www.youtube.com/c/SmartProgramming)

ANDROID APP : Smart Programming
(https://play.google.com/store/apps/details?id=com.smartprogramming)

https://www.facebook.com/smartprogramming.india

https://www.instagram.com/smart_programming

Java
android
python
Data Science
Machine Learning
Artificial Intelligence
M E A N
Full Stack
BIG DATA
hadoop

Constructors in Java

=> Constructors :-

-> WHAT IS CONSTRUCTOR : Constructors are the special methods having same name as that of class name and does not have any return type

-> EXAMPLE :-

```
class Animal
{
	Animal()
	{


	}
}
```

-> USE OF CONSTRUCTOR :- Constructors are used to initialize an object but not for object creation

-> WHEN CONSTRUCTORS ARE EXECUTED :- Constructors are executed exactly at the time of object creation, not before or after object creation

-> HOW CONSTRUCTORS ARE EXECUTED :- Constructors are executed automatically when we create an object

-> SYNTAX :-

```
access-modifiers ClassName(list of parameters) throws Exception1, Exception2, --
{
	//initialization code
}
```

-> We can use any access-modifier for the constructor i.e. public, private, protected or default. This is done to control the object creation

-> We cannot use abstract, final, static, synchronized etc keywords with constructors

-> TYPES OF CONSTRUCTORS :- There are 3 types of constructors :-

1. Default Constructors (compiler)
2. 0-Argument Constructors (programmer)
3. Parametrized Constructors (programmer)

---

1. Default Constructors :-

-> Whenever we dont create any constructor in class, then compiler will always create a constructor which is known as default constructor

-> Default constructors are used to provide the default values to the objects like 0, null etc depending on the type.

-> Note : If programmer creates any one constructor then compiler will not generate default constructor

-> Prototype of default constructor :-

1. Access-modifier of default constructor will be same at that of class access-modifier

2. Access-modifier of default constructor cannot be private or protected because outer class canot be private or protected

3. Default constructor has only one line of code i.e. super();

2. 0-Argument Constructors :-

-> These constructors are created by the programmer

```
class Test
{
        Test()
        {


        }
}
```

3. Parametrized Constructors :-

-> These constructors are created by the programmer

```
class Test
{
    Test(int a, int b)
    {


    }
}
```

=> What is difference between Methods & Constructors :-

1. Methods always have return type

Constructors does not have any return type even void

2. Methods can have any valid name

Constructors always have same name as that of class name

3. Methods are used to perform any particular task

Constructors are always used to initialize an object

4. We have to call the methods explicitly by using object name or class name

Constructors are called automatically when we create an object

5. If we dont create any method then compiler will not generate any method

If we dont create any constructor then compiler will generate default constructor

---

=> Topics related to constructor :-

1. Constructors with inheritance
2. Constructors overloading and overriding
3. Constructors chaining (using this keyword)
4. Use of super keyword with constructor
5. Constructors with abstract class & interface
6. Constructors with exception handling
7. Copy constructor

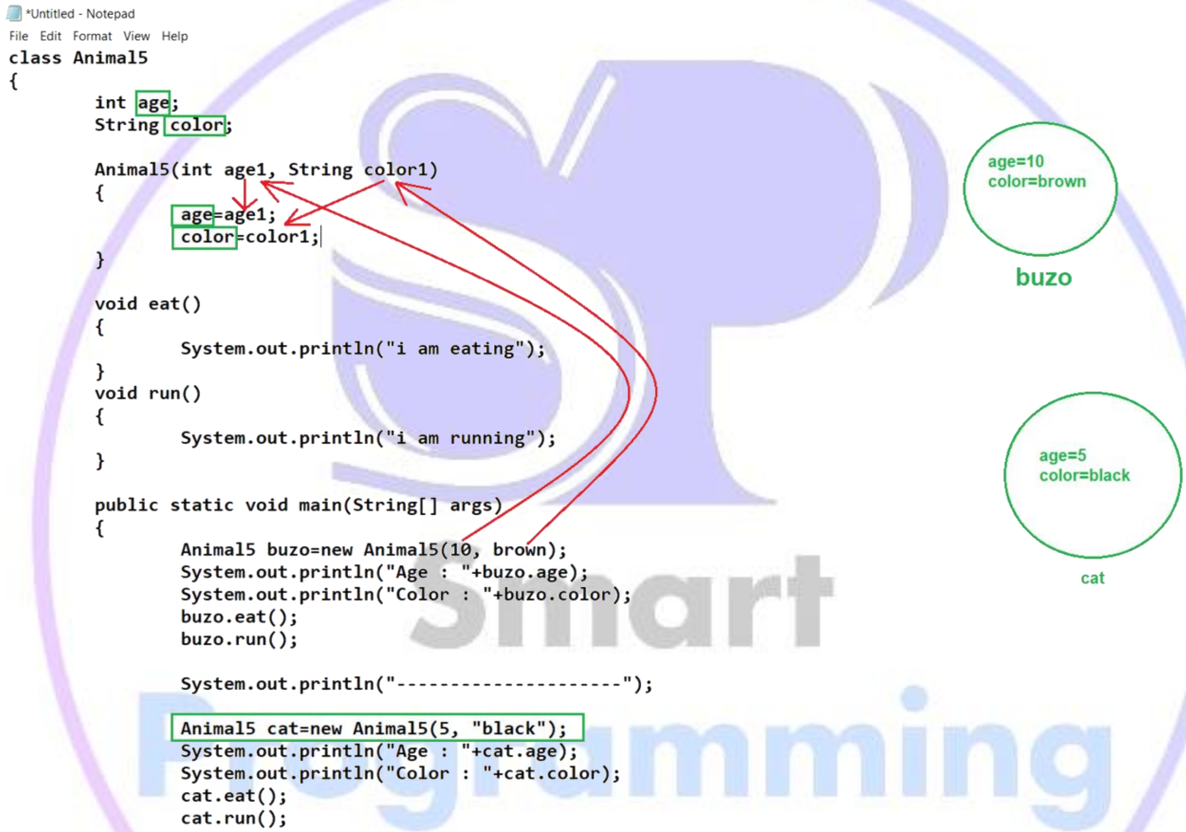
=> NOTE :-

-> Constructor is predefined class present in java.lang.reflect package

-> This Constructor class is used to get constructor related information

---

⇨ Program Flow


*Untitled - Notepad
File Edit Format View Help
age=10
color=brown
buzo
age=5
color=black
cat

```
class Animal5
{
        int age;
        String color;

        Animal5(int age1, String color1)
        {
                age=age1;
                color=color1;
        }

        void eat()
        {
                System.out.println("i am eating");
        }
        void run()
        {
                System.out.println("i am running");
        }

        public static void main(String[] args)
        {
                Animal5 buzo=new Animal5(10, brown);
                System.out.println("Age : "+buzo.age);
                System.out.println("Color : "+buzo.color);
                buzo.eat();
                buzo.run();

                System.out.println("--------------------");

                Animal5 cat=new Animal5(5, "black");
                System.out.println("Age : "+cat.age);
                System.out.println("Color : "+cat.color);
                cat.eat();
                cat.run();
```

Company Links & Contacts

Company Name: Smart Programming (+91 62838-30308)

Address : Chandigarh & Mohali (Punjab), India

Websites: https://www.smartprogramming.in/

https://courses.smartprogramming.in

Android App: https://play.google.com/store/apps/details?id=com.smartprogramming

YouTube Channel: https://www.youtube.com/c/SmartProgramming

Smart Programming : YouTube Channel

An investment in Knowledge pays the best interest....

Smart Programming
We Educate - We Develop

+91 62838-30308
Call us to Learn
Latest Technologies

City : Mohali (Punjab),
& Chandigarh
(India)

WEBSITE : http://www.smartprogramming.in

BUY COURSES ON : https://courses.smartprogramming.in

YOUTUBE CHANNEL : Smart Programming (https://www.youtube.com/c/SmartProgramming)

ANDROID APP : Smart Programming
(https://play.google.com/store/apps/details?id=com.smartprogramming)

https://www.facebook.com/smartprogramming.india

https://www.instagram.com/smart_programming

Java
android
python
Data Science
Machine Learning
Artificial Intelligence
M E A N
Full Stack
BIG DATA
hadoop

Object Creation Explanation in Java

=> How Objects are created :-

---

1. When We compile the program i.e. javac AnimalMain.java

Compiler will check the syntax and if syntax is correct then it will generate .class files (no of .class files generated depends on the no of classes we have created)

2. When we run the program i.e. java AnimalMain

2.1 AnimalMain.class file will be loaded in JVM memory area i.e. in Method Area

2.2 An object of java.lang.Class class will be created in Heap Area in which AnimalMain class metadata will be stored

2.3 Now Main Method will execute and for this JVM will create a new thread known as main thread

2.4 As soon as main thread is created then JVM will create main stack in stack area

2.5 Now first line in main method will execute i.e. Animal ob=new Animal(); now Animal.class file will load in method area and a new object of java.lang.Class class is created in heap area in which metadata of Animal class will be stored

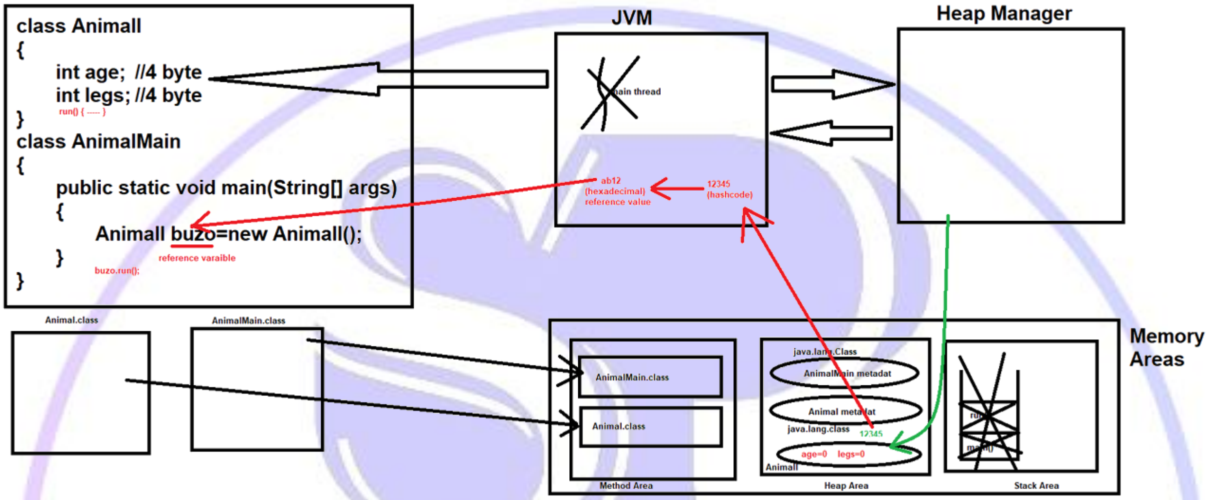
2.6 Now there is new keyword so new object creation process will start. JVM will instruct heap manager to create an object of Animal class but heap manager will ask for object size to JVM. Then JVM will calculate the size of object according to the number of instance variables that are declared in Animal class and this size will be taken by heap manager and heap manager will create an object in heap area

2.7 As soon as heap manager creates an object, a unique integer value will be assigned to the object which is known as "hashcode"

2.8 This hascode value will be provided to the JVM and JVM will convert this hascode vallue into hexadecimal form and this hexadecimal value is known as reference value

2.9. Now this hexadecimal value will be assigned to the variable which is known as reference variable

2.10 Now object will be initialized that means all instance variables will be assigned by default values or by their original values



```
class Animall
{
      int age;  //4 byte
      int legs; //4 byte
      run() { ----- }
}
class AnimalMain
{
      public static void main(String[] args)
      {
            Animall buzo=new Animall();
      }
            buzo.run();
}
```

reference varaible
Animal.class
AnimalMain.class
JVM
main thread
ab12
(hexadecimal)
reference value
12345
(hashcode)
Heap Manager
Memory Areas
AnimalMain.class
Animal.class
Method Area
java.lang.Class
AnimalMain metadat
Animal metadat
java.lang.class
12345
age=0 legs=0
Animall
Heap Area
run()
main()
Stack Area

=> java.lang.Class :- It represents the classes and interfaces which are used in running java application

=> java.lang.Object :- This is the parent class of all the classes in java.

When we create any class than that class will inherit Object class either directly or indirectly

=> java.lang.reflect.Method

=> java.lang.reflect.Constructor

Company Links & Contacts

Company Name: Smart Programming (+91 62838-30308)

Address : Chandigarh & Mohali (Punjab), India

Websites: https://www.smartprogramming.in/

https://courses.smartprogramming.in

Android App: https://play.google.com/store/apps/details?id=com.smartprogramming

YouTube Channel: https://www.youtube.com/c/SmartProgramming

Smart Programming : YouTube Channel

An investment in Knowledge pays the best interest….

Smart Programming
We Educate - We Develop

☎ +91 62838-30308
Call us to Learn Latest Technologies
City : Mohali (Punjab), & Chandigarh (India)

WEBSITE : http://www.smartprogramming.in

BUY COURSES ON : https://courses.smartprogramming.in

YOUTUBE CHANNEL : Smart Programming (https://www.youtube.com/c/SmartProgramming)

ANDROID APP : Smart Programming (https://play.google.com/store/apps/details?id=com.smartprogramming)

https://www.facebook.com/smartprogramming.india

https://www.instagram.com/smart_programming

Java
android
python
Data Science
Machine Learning
Artificial Intelligence
M E A N
Full Stack
BIG DATA
hadoop

Inheritance in Java

Relationship Between Java Classes :-

=> Use of relationship between java classes :-

1. Code Reusability
2. Less Execution Time
3. Less Memory Usage

=> Types of relationships :-

1. IS-A Relationship (Inheritance)

-> IS-A relationship is one in which data members of one class is obtained into another class through the concept of inheritance

-> Types of IS-A relationship :-

1.1 Single Inheritance

1.2 Multilevel Inheritance

1.3 Hierarchical Inheritance

1.4 Multiple Inheritance

1.5 Hybrid Inheritance

2. HAS-A Relationship (Association)

-> HAS-A relationship is one in which an object of one class is created as a data member into another class.

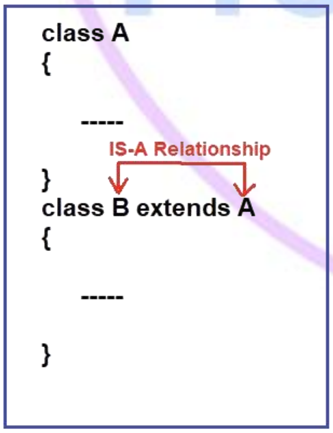
-> Types of HAS-A relationship :-

2.1 Aggregation

2.2 Composition

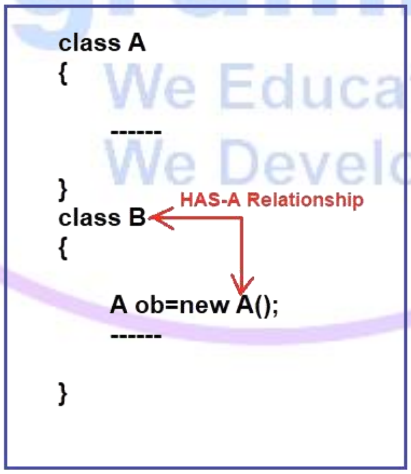
3. USES-A Relationship (Dependence)

-> USES-A relationship is one in which a method of one class is using an object of another class



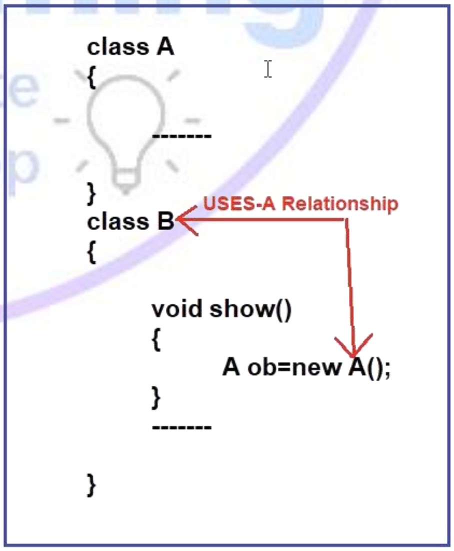
```
class A
{
    -----
}
class B extends A
{
    -----
}
```

IS-A Relationship



```
class A
{
    ------
}
class B
{
    A ob=new A();
    ------
}
```

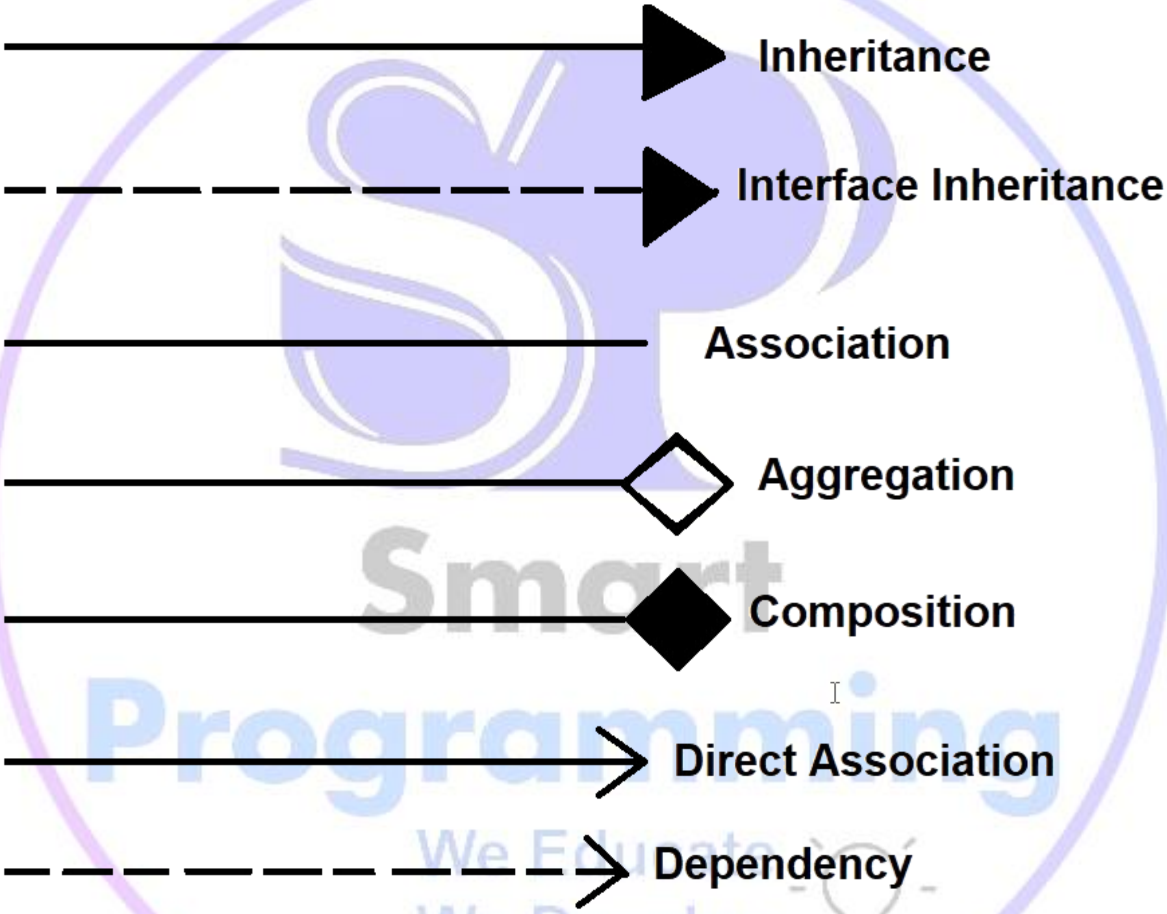
HAS-A Relationship



```
class A
{
    -------
}
class B
{
    void show()
    {
        A ob=new A();
    }
    -------
}
```

USES-A Relationship

=> UML Notation in Java :-


Inheritance
Interface Inheritance
Association
Aggregation
Composition
Direct Association
Dependency

=> Terms used for inheritance :-

1. Class
2. Sub-Class / Child Class
3. Super-Class / Parent Class
4. Reusability

=> IS-A Relationship :-

-> It is also known as "Inheritance"

-> IS-A Relationship or Inheritance is achieved by using "extends" keyword

-> All java classes except Object class will always have one parent class thus we can say that the total java API is implemented based on inheritance concept

=> Use of inheritance :-

1. Code Reusability
2. For Method Overriding to achieve runtime polymorphism

=> Syntax of inheritance :-

```
class Sub-Class extends Super-Class
{
	//body
}
```

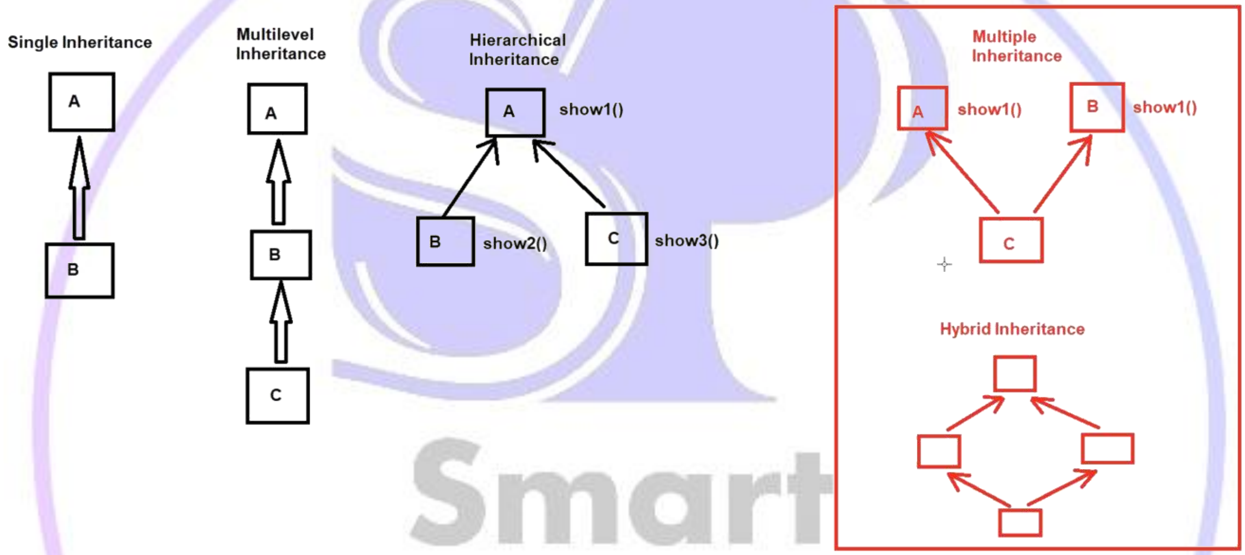
=> Types of Inheritance :-

-> Total there are 5 types of inheritance :-

1. Single Inheritance
2. Multilevel Inheritance
3. Hierarchical Inheritance

4. Multiple Inheritance

5. Hybrid Inheritance


Single Inheritance
A
B
Multilevel Inheritance
A
B
C
Hierarchical Inheritance
A
show1()
B
show2()
C
show3()
Multiple Inheritance
A
show1()
B
show1()
C
Hybrid Inheritance

=> Points to Remember :-

-> Default Parent Class :- By default if any java class does not inherit any parent class then it inherits Object class

-> Parent class can only be one :- There can be only one parent class for every class and due to this java does not support multiple inheritance

-> Which part is not inherited :-

1. Private members of parent class is not inherited in child class

2. Constructors are not inherited because constructors are not the part of class members (class members are only fields, methods, nested classes)

-> Cyclic inheritance is not possible

1. class A extends A

2. class A extends B { - }

class B extends A { - }

-> Multiple and Hybrid inheritance is not possible in case of classes but it is possible in case of interfaces

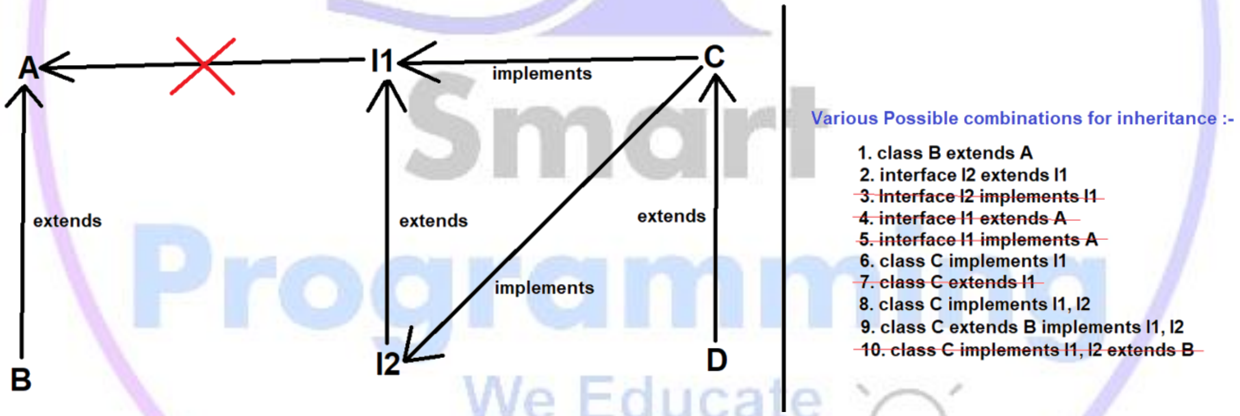
---

Interview Questions :-

1. Why java does not support multiple inheritance

2. How we can achieve Multiple and Hybrid inheritance

3. Various possible combinations for inheritance



Various Possible combinations for inheritance :-

1. class B extends A
2. interface I2 extends I1
3. ~~Interface I2 implements I1~~
4. ~~interface I1 extends A~~
5. ~~interface I1 implements A~~
6. class C implements I1
7. ~~class C extends I1~~
8. class C implements I1, I2
9. class C extends B implements I1, I2
10. ~~class C implements I1, I2 extends B~~

Company Links & Contacts

Company Name: Smart Programming (+91 62838-30308)

Address : Chandigarh & Mohali (Punjab), India

Websites: https://www.smartprogramming.in/

https://courses.smartprogramming.in

Android App:
https://play.google.com/store/apps/details?id=com.smartprogramming

YouTube Channel:
https://www.youtube.com/c/SmartProgramming

Smart Programming : YouTube Channel

An investment in Knowledge pays the best interest….

Smart Programming
We Educate - We Develop

☎ +91 62838-30308
Call us to Learn
Latest Technologies

City : Mohali (Punjab), & Chandigarh (India)

WEBSITE : http://www.smartprogramming.in

BUY COURSES ON : https://courses.smartprogramming.in

YOUTUBE CHANNEL : Smart Programming (https://www.youtube.com/c/SmartProgramming)

ANDROID APP : Smart Programming
(https://play.google.com/store/apps/details?id=com.smartprogramming)

https://www.facebook.com/smartprogramming.india

https://www.instagram.com/smart_programming

Java
android
python
Data Science
Machine Learning
Artificial Intelligence
M E A N
Full Stack
BIG DATA
hadoop

Association in Java

=> HAS-A Relationship (Association) :-

-> Association is relation between two separate classes which establishes through their objects

-> Association has 4 types :-

1. One-to-One Association
2. One-to-Many Association
3. Many-to-One Association
4. Many-to-Many Association

-> Association has 2 forms :-

1. Composition
2. Aggregation

=> One-to-One Association :-

-> Real World Example :-

1. One person has one passport
2. One employee has one employee id
3. One student has one roll no

-> It is a relation between entities where one instance of an entity should be mapped only to one instance of another entity

---

=> One-to-Many Association :-

-> Real World Example :-

1. One person can have multiple phone numbers
2. One student can have multiple courses

3. One person can have multiple bank accounts

4. One machine can have multiple functionalities like tea, coffee etc

5. One customer can have multiple orders

6. One class can have multiple students

-> It is the relation between entity classes where one instance of an entity should be mapped with multiple instance of another entity

---

=> Many-to-One Association :-

-> Real World Example :-

1. Many colleges can associate with single university

2. Multiple cities exist in one state

3. Many ports can exist in one extention

4. Multiple students can have one branch

-> It is a relation between entities where multiple instances of an entity should be mapped with exactly one instance of another entity

---

=> Many-to-Many Association

-> Real World Example :-

1. Multiple customers can buy multiple products and multiple products can be bought by multiple customers

2. Multiple students can associate with multiple teachers and multiple teachers are associated with multiple students

3. Multiple students can learn multiple languages

-> It is relation between entities where multiple instances of an entity should be mapped with multiple instances of another entity

Company Links & Contacts

Company Name: Smart Programming (+91 62838-30308)

Address : Chandigarh & Mohali (Punjab), India

Websites: https://www.smartprogramming.in/

https://courses.smartprogramming.in

Android App: https://play.google.com/store/apps/details?id=com.smartprogramming

YouTube Channel: https://www.youtube.com/c/SmartProgramming

Smart Programming : YouTube Channel

An investment in Knowledge pays the best interest….

Smart Programming
We Educate - We Develop

☎ +91 62838-30308
Call us to Learn
Latest Technologies

City : Mohali (Punjab),
& Chandigarh
(India)

WEBSITE : http://www.smartprogramming.in

BUY COURSES ON : https://courses.smartprogramming.in

YOUTUBE CHANNEL : Smart Programming (https://www.youtube.com/c/SmartProgramming)

ANDROID APP : Smart Programming
(https://play.google.com/store/apps/details?id=com.smartprogramming)

https://www.facebook.com/smartprogramming.india

https://www.instagram.com/smart_programming

Java
ANDROID
python
Data Science
Machine Learning
Artificial Intelligence
M E A N
Full Stack
BIG DATA
hadoop

Association in Java

=> HAS-A Relationship (Association) :-

-> Association is relation between two separate classes which establishes through their objects

-> Association has 4 types :-

1. One-to-One Association
2. One-to-Many Association
3. Many-to-One Association
4. Many-to-Many Association

-> Association has 2 forms :-

1. Composition
2. Aggregation

=> One-to-One Association :-

-> Real World Example :-

1. One person has one passport
2. One employee has one employee id
3. One student has one roll no

-> It is a relation between entities where one instance of an entity should be mapped only to one instance of another entity

---

=> One-to-Many Association :-

-> Real World Example :-

1. One person can have multiple phone numbers
2. One student can have multiple courses

3. One person can have multiple bank accounts

4. One machine can have multiple functionalities like tea, coffee etc

5. One customer can have multiple orders

6. One class can have multiple students

-> It is the relation between entity classes where one instance of an entity should be mapped with multiple instance of another entity

---

=> Many-to-One Association :-

-> Real World Example :-

1. Many colleges can associate with single university

2. Multiple cities exist in one state

3. Many ports can exist in one extention

4. Multiple students can have one branch

-> It is a relation between entities where multiple instances of an entity should be mapped with exactly one instance of another entity

---

=> Many-to-Many Association

-> Real World Example :-

1. Multiple customers can buy multiple products and multiple products can be bought by multiple customers

2. Multiple students can associate with multiple teachers and multiple teachers are associated with multiple students

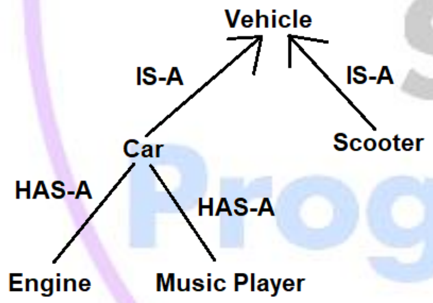
3. Multiple students can learn multiple languages

-> It is relation between entities where multiple instances of an entity should be mapped with multiple instances of another entity

---

-> Association has 2 forms :-

1. Composition (strong bonding)
2. Aggregation (week bonding)

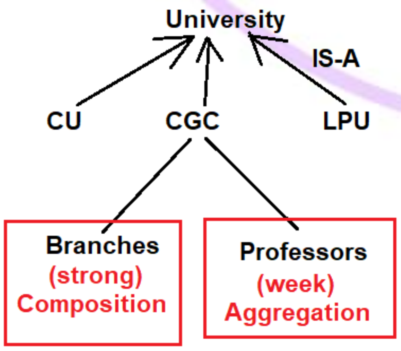

Vehicle
IS-A
IS-A
Car
Scooter
HAS-A
HAS-A
Engine
Music Player

Car IS-A Vehicle

Scooter IS-A Vehicle

Car HAS-A Engine (Strong Bonding - Composition)

Car HAS-A MusicPlayer (Week Bonding - Aggregation)


University
IS-A
CU
CGC
LPU
Branches
(strong)
Composition
Professors
(week)
Aggregation

=> Difference between Composition & Aggregation :-

1. Real World Example :-

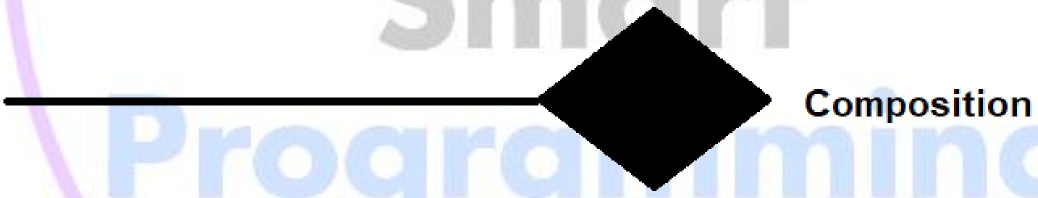
    Composition :- Car HAS-A Engine

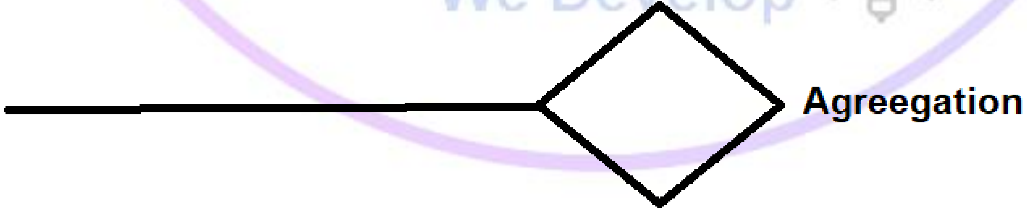
    Aggregation :- Car HAS-A MusicPlayer

2. UML Diagram :-

    Composition :- Car ------<filled> Engine

    Aggregation :- Car ------<> MusicPlayer





3. Composition :- part-of

Aggregation :- has

4. Type of bonding :-

Composition :- Strong Bonding

Aggregation :- Week Bonding

=> Where we have to use IS-A relation and HAS-A relation

If we want to inherit all the properties of parent class into child class then we can use IS-A relation but if we want only partial inherit then we have to use HAS-A relation

---

Company Links & Contacts

Company Name: Smart Programming (+91 62838-30308)

Address : Chandigarh & Mohali (Punjab), India

Websites: https://www.smartprogramming.in/

https://courses.smartprogramming.in

Android App: https://play.google.com/store/apps/details?id=com.smartprogramming

YouTube Channel: https://www.youtube.com/c/SmartProgramming

Smart Programming : YouTube Channel

An investment in Knowledge pays the best interest....

Smart Programming
We Educate - We Develop

☎ +91 62838-30308
Call us to Learn
Latest Technologies

City : Mohali (Punjab),
& Chandigarh
(India)

WEBSITE : http://www.smartprogramming.in

BUY COURSES ON : https://courses.smartprogramming.in

YOUTUBE CHANNEL : Smart Programming (https://www.youtube.com/c/SmartProgramming)

ANDROID APP : Smart Programming
(https://play.google.com/store/apps/details?id=com.smartprogramming)

https://www.facebook.com/smartprogramming.india

https://www.instagram.com/smart_programming

Java
android
python
Data Science
Machine Learning
Artificial Intelligence
M E A N
Full Stack
BIG DATA
hadoop

Dependency Injection & USES-A Relation in Java

=> Dependency Injection :-

-> DI is the concept in which objects gets the other required objects from outside entity

-> The process of injecting or inserting dependent(contained) object into the container object is known as Dependency Injection

-> NEED :- While programming, java classes should be as minimum dependent as possible on each other. By this we will increase the possibility of reusing these classes and to be able to test them independently

-> TYPES OF DEPENDENCY INJECTION :-

1. Constructor Dependency Injection
2. Setter Method Dependency Injection

=> Circular Dependency Injection :-

-> It is the process of depending two objects in circular form i.e. first object is dependent on second object and second object is dependent on first object

-> Circular DI can be achieved by setter method dependency injection but not in case of constructor dependency injection

=> USES-A Relationship (Dependence)

-> USES-A relationship is one in which a method of one class is using an object of another class

-> For example : Transaction Uses Account to deposit() the amount

Company Links & Contacts

Company Name: Smart Programming (+91 62838-30308)

Address : Chandigarh & Mohali (Punjab), India

Websites: https://www.smartprogramming.in/

https://courses.smartprogramming.in

Android App:
https://play.google.com/store/apps/details?id=com.smartprogramming

YouTube Channel:
https://www.youtube.com/c/SmartProgramming

Smart Programming : YouTube Channel

An investment in Knowledge pays the best interest….

Smart Programming

We Educate - We Develop

☎ +91 62838-30308

Call us to Learn Latest Technologies

City : Mohali (Punjab), & Chandigarh (India)

WEBSITE : http://www.smartprogramming.in

BUY COURSES ON : https://courses.smartprogramming.in

YOUTUBE CHANNEL : Smart Programming (https://www.youtube.com/c/SmartProgramming)

ANDROID APP : Smart Programming (https://play.google.com/store/apps/details?id=com.smartprogramming)

https://www.facebook.com/smartprogramming.india

https://www.instagram.com/smart_programming

Java
android
python
Data Science
Machine Learning
Artificial Intelligence
M E A N
Full Stack
BIG DATA
hadoop

Polymorphism (Method Overloading) in Java

=> Polymorphism :-

-> Poly (many) + Morhism (forms, structure)

-> Real world example : sound, water, many brand clothes in one shop, person, single institute has multiple trainer etc

-> Advantage :- It provides the flexibility to develop an application i.e. it allows us to perform a single task by different ways.

-> Types of Polymorphism :-

1. Compile Time Polymorphism
2. Runtime Polymorphism

=> Compile Time Polymorphism :-

-> It is also known as Static Polymorphism or Early Binding

-> If the polymorphism is achieved at compile time then it is compile time polymorphism

-> Compile Time Polymorphism can be achieved by 2 ways :-

1. Method Overloading
2. Operator Overloading (is not supported in java except + symbol)

=> Runtime Polymorphism :-

-> It is also known as Dynamic Polymorphism or Late Binding

-> If the polymorphism is existed at runtime then it is known as Runtime Polymorphism

-> Runtime Polymorphism can be achieved by "Method Overriding"

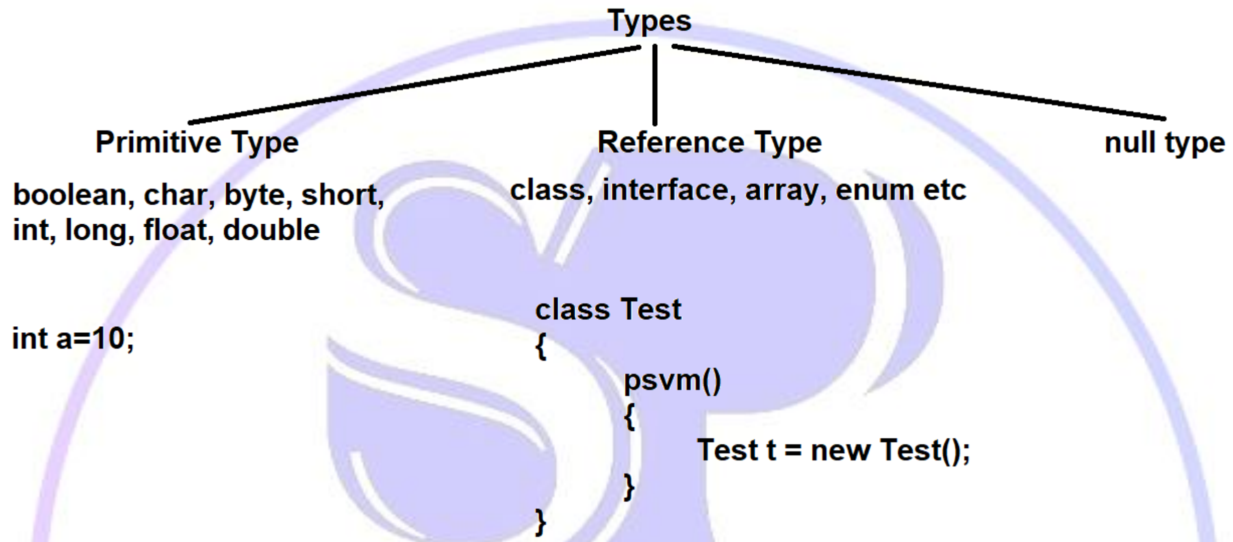
=> Method Overloading :-

-> The process of compiler trying to resolve the method call based on reference type is known as method overloading

-> Rules for method overloading :-

1. Same name
2. Within same class
3. Different parameters
    - -> No of parameters
    - -> Type of parameters
    - -> Sequence of parameters

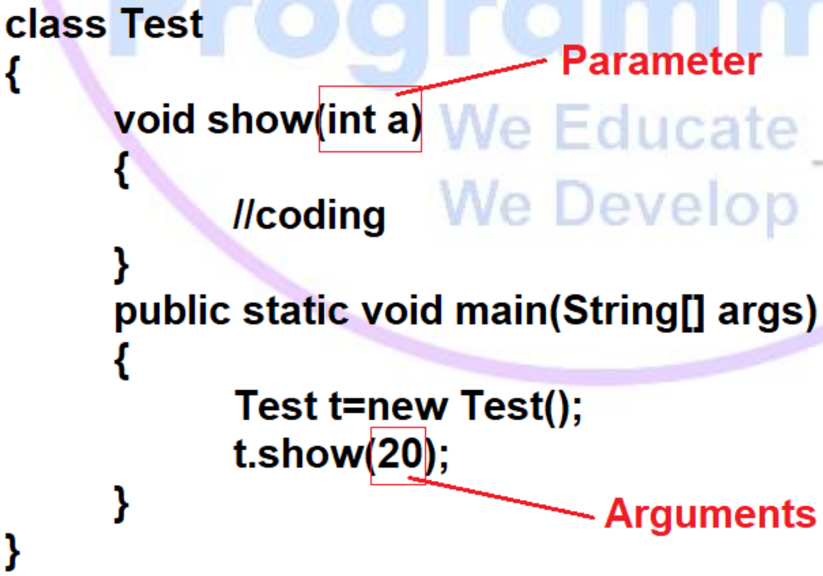
What is reference type ?


Types
Primitive Type
Reference Type
null type
boolean, char, byte, short,
int, long, float, double
class, interface, array, enum etc

```
int a=10;
```

```
class Test
{
        psvm()
        {
                Test t = new Test();
        }
}
```

What is parameter and argument ?


Parameter
Arguments

```
class Test
{
        void show(int a)
        {
                //coding
        }
        public static void main(String[] args)
        {
                Test t=new Test();
                t.show(20);
        }
}
```

Interview Questions :-

1. Operator Overloading is not supported in java but + operator is overloaded

-> Operator Overloading concept is achieved only by java designers but it cannot be achieved by developers like us.

2. What is difference between Compile Time Polymorphism & Runtime Polymorphism

3. What is varargs

4. Can we overload main method -> Yes

5. Can we overload constructors -> Yes

Company Links & Contacts

Company Name: Smart Programming (+91 62838-30308)

Address : Chandigarh & Mohali (Punjab), India

Websites: https://www.smartprogramming.in/

https://courses.smartprogramming.in

Android App:
https://play.google.com/store/apps/details?id=com.smartprogramming

YouTube Channel:
https://www.youtube.com/c/SmartProgramming

Smart Programming : YouTube Channel

An investment in Knowledge pays the best interest....

Smart Programming

We Educate - We Develop

+91 62838-30308

Call us to Learn Latest Technologies

City : Mohali (Punjab), & Chandigarh (India)

WEBSITE : http://www.smartprogramming.in

BUY COURSES ON : https://courses.smartprogramming.in

YOUTUBE CHANNEL : Smart Programming (https://www.youtube.com/c/SmartProgramming)

ANDROID APP : Smart Programming (https://play.google.com/store/apps/details?id=com.smartprogramming)

https://www.facebook.com/smartprogramming.india

https://www.instagram.com/smart_programming

Java
android
python
Data Science
Machine Learning
Artificial Intelligence
M E A N
Full Stack
BIG DATA
hadoop

Polymorphism (Method Overriding) in Java

=> Method Overriding :-

-> The process of JVM trying to resolve the method call based on reference type is known as method overriding

-> Overriding is the feature by which child class trying to change the implementation of parent class method

-> Rules for method overriding :-

1. Same name
2. Within different class
3. Same parameters
    - -> No of parameters
    - -> Type of parameters
    - -> Sequence of parameters
4. IS-A relationship

-> Cases for method overriding :

1. If we change the return type in method overriding then it will provide compile time error
2. We can provide child class as a return type for overriding method and this concept is known as covarent return type
3. Child class method should have equal or higher access modifier as compared to parent method access modifier in method overriding
4. We cannot override private, final and static methods
5. We cannot override constructors
6. We cannot override main method

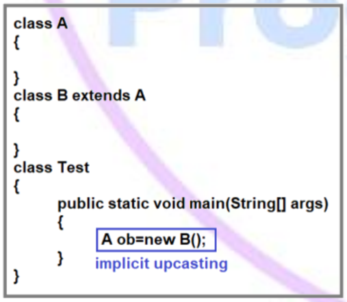
=> Typecasting : The process of converting one data type into another is known as typecasting

=> Object Typecasting :

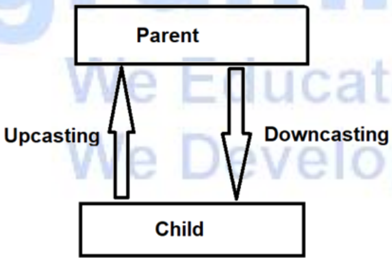
-> The process of converting one object into another object is known as Object Typecasting

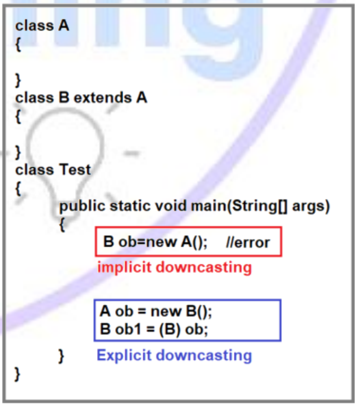
-> Object typecasting is of 2 types :-

1. Upcasting
2. Downcasting



```
class A
{

}
class B extends A
{

}
class Test
{
        public static void main(String[] args)
        {
                A ob=new B();
        }       implicit upcasting
}
```





```
class A
{

}
class B extends A
{

}
class Test
{
        public static void main(String[] args)
        {
                B ob=new A();    //error
                implicit downcasting


                A ob = new B();
                B ob1 = (B) ob;
        }       Explicit downcasting
}
```

Interview Questions :-

1. What is difference between method overloading & method overriding

2. What is upcasting & downcasting ?

-> Upcasting : Object typecasting in which child object is typecasted into parent object

Downcasting : Object typecasting which parent object is typecasted into child object

-> Upcasting : Implicit upcasting is possible

Downcasting : Implicit downcasting is not possible but forcefully we can do i.e. explicit downcasting is possible

3. What is covarent return type ?

-> Before JDK 1.5 version, covarent return type concept was not there

-> But in JDK 1.5 version and above version we can provide any child class as a return type for overriding method

4. Which methods cannot override ?

-> private methods cannot be override

-> final methods cannot be override

-> static methods cannot be override (method hiding concept)

Company Links & Contacts

Company Name: Smart Programming (+91 62838-30308)

Address : Chandigarh & Mohali (Punjab), India

Websites: https://www.smartprogramming.in/

https://courses.smartprogramming.in

Android App: https://play.google.com/store/apps/details?id=com.smartprogramming

YouTube Channel: https://www.youtube.com/c/SmartProgramming

Smart Programming : YouTube Channel

An investment in Knowledge pays the best interest….

Smart Programming
We Educate - We Develop

WEBSITE : http://www.smartprogramming.in

BUY COURSES ON : https://courses.smartprogramming.in

YOUTUBE CHANNEL : Smart Programming (https://www.youtube.com/c/SmartProgramming)

ANDROID APP : Smart Programming
(https://play.google.com/store/apps/details?id=com.smartprogramming)

https://www.facebook.com/smartprogramming.india

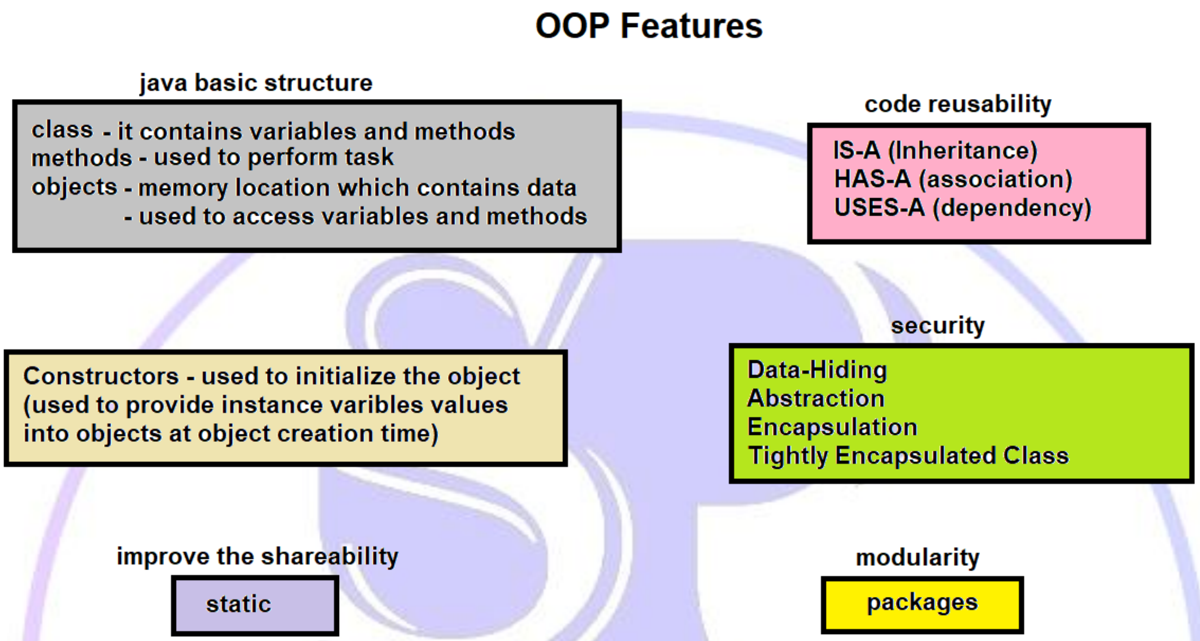
https://www.instagram.com/smart_programming

+91 62838-30308
Call us to Learn
Latest Technologies

City : Mohali (Punjab),
& Chandigarh
(India)

Java
android
python
Data Science
Machine Learning
Artificial Intelligence
M E A N
Full Stack
BIG DATA
hadoop

Abstraction in Java



OOP Features

java basic structure

class - it contains variables and methods
methods - used to perform task
objects - memory location which contains data
- used to access variables and methods

code reusability

IS-A (Inheritance)
HAS-A (association)
USES-A (dependency)

Constructors - used to initialize the object
(used to provide instance varibles values
into objects at object creation time)

security

Data-Hiding
Abstraction
Encapsulation
Tightly Encapsulated Class

improve the shareability

static

modularity

packages

=> Data-Hiding :-

-> Data-Hiding is the process of hiding the data from outside users

-> It is achieved by private access-modifier

-> Example :

```
class Account
{
```

```
                  private int balance;       //data hiding
                  //code---
          }
```

-> To access or modify or update private variables java provide some special methods i.e. getter and setter methods

-> It is highly recomended to declare variables (data members) as private

---

=> Abstraction :-

-> Abstraction is hiding the details (hiding the implementation part) and just highlight the main services

-> Real world example : Car (internal working of breaks, gears etc are hidden from user), Bank (internal working of credit/debit is hided form user and only main component i.e. textfield and button is shown) etc

-> It is achieved by abstract class & interfaces

=> Abstract class :-

-> Abstract class are those which can contain both concrete methods and abstract methods

-> Syntax :

```
abstract class Test

{
        //concrete methods
        void sum()
        {
                //coding
                }
                //abstract methods
                abstract void show();
            }
```

=> Abstract method :-

-> For example : Devils

-> 1. Abstract methods are those whose implementation part is hided

2. Abstract methods are those which does not have body or implementation part

3. Abstract methods are those which have only declaration part, not implementation

-> Syntax :

```
abstract void sum();
```

-> To declare abstract methods we have to use "abstract" keyword

=> Points to remember :-

1. If any class contains abstract methods then that class should be declared as abstract class
2. If we declare any abstract class, then it can contain both concrete methods as well as abstract methods and it can contain only concrete methods
3. We cannot create an object of abstract class or we cannot instantiate an abstract class but we can declare reference for an abstract class
4. If any class inherits abstract class then it should implement all the abstract methods or that class should also be declared as abstract
5. Whenever we use abstraction concept we are using method overriding concept also
6. Abstract class can have constructors
7. Abstract class can inherit concrete class
8. We can overload abstract methods

Interview Questions :-

1. What is difference between concrete methods and abstract methods ?

-> Concrete Methods : which have both declaration and implementation part

Abstract Methods : which have only declaration part

-> Concrete Methods : we dont need to use any keyword

Abstract Methods : we have to use "abstract" keyword

-> Concrete Methods : provides less shareability

Abstract Methods : provides more shareability

-> Concrete Methods : can be provided in concrete class and abstract class

Abstract Methods : can be provided only in abstract class

2. What is difference between concrete class and abstract class ?

-> Concrete Class : which can contain only concrete methods

Abstract Class : which can contain both concrete methods as well as abstract methods

-> Concrete Class : we can create an object of concrete class

Abstract Class : we cannot create an object of abstract class

-> Concrete Class : it does not require any specified keyword

Abstract Class : it can be declared only by using "abstract" keyword

Company Links & Contacts

Company Name: Smart Programming (+91 62838-30308)

Address : Chandigarh & Mohali (Punjab), India

Websites: https://www.smartprogramming.in/

https://courses.smartprogramming.in

Android App: https://play.google.com/store/apps/details?id=com.smartprogramming

YouTube Channel: https://www.youtube.com/c/SmartProgramming

Smart Programming : YouTube Channel

An investment in Knowledge pays the best interest....

Smart Programming
We Educate - We Develop
+91 62838-30308
Call us to Learn
Latest Technologies
City : Mohali (Punjab),
& Chandigarh
(India)
WEBSITE : http://www.smartprogramming.in
BUY COURSES ON : https://courses.smartprogramming.in
YOUTUBE CHANNEL : Smart Programming (https://www.youtube.com/c/SmartProgramming)
ANDROID APP : Smart Programming
(https://play.google.com/store/apps/details?id=com.smartprogramming)
https://www.facebook.com/smartprogramming.india
https://www.instagram.com/smart_programming

Java
android
python
Data Science
Machine Learning
Artificial Intelligence
M E A N
Full Stack
BIG DATA
hadoop

Interfaces in Java

=> Interface

-> Interfaces are similar to abstract class which can contain variables and methods but having all the variables as "public static final" and methods as "public abstract"

-> In simple way we can say that all the methods in an interface are abstract and thus methods cannot have implementation part or body part

-> Interface is a blueprint of class which specifies what must do and not how

-> Syntax :

```
access-modifier interface InterfaceName extends InterfaceName, -, -
{
    //variables (public static final)
    //abstract methods (public abstract)
}
```

```
interface Vehicle
{
        void start();
        void changeGear();
}
```

-> Use of Interfaces :-

1. It is used to achieve total abstraction
2. It is used to achieve multiple inheritance (as multiple inheritance is not supported in java in case of classes)
3. It is used to achieve loose coupling

Interview Questions :-

1. What is difference between concrete class, abstract class & interface

-> Concrete Class : Syntax : class ClassName{}

Abstract Class : Syntax : abstract class ClassName{}

Interface : Syntax : interface InterfaceName{}

-> Concrete Class : we can declare only concrete methods

Abstract Class : we can declare concrete methods and abstract methods

Interface : we can declare only abstract methods

-> Concrete Class : we can create objects

Abstract Class : We cannot create an object but we can declare reference variable name

Interface : We cannot create an object but we can declare reference variable name

-> Concrete Class : we cannot achieve abstraction

Abstract Class : we can achieve partial abstraction

Interface : we can achieve full abstraction

-> Concrete Class : methods & variables are same as we declare

Abstract Class : methods & variables are same as we declare

Interface : methods we have declared are always "public abstract" and variables we have declared are always "public static final"

-> Concrete Class : constructors are allowed

Abstract Class : constructors are allowed

Interface : constructors are not allowed

-> Concrete Class : multiple inheritance is not supported

Abstract class : multiple inheritance is not supported

Interface : multiple inheritance is supported

=> Points to remember :

1. Interfaces cannot be private or protected but nested interface can be anything i.e. private, protected, default & public

2. Interface represents IS-A relationship

3. Interface new features :-

3.1 (JDK 8) We can create default methods in an interface which have implementation part

3.2 (JDK 8) We can create static methods in an interface

3.3 (JDK 9) We can create private methods in an interface

3.4 (JDK 9) We can create private static methods in an interface

=> What is marker interface ?

-> Any interface which does not contain any abstract method or any variable is known as marker interface

-> It is used to provide some extra feature or abilities to the object at runtime

-> For example :

1. Cloneable interface (java.lang)
2. Serializable interface (java.io)
3. Remote interface (java.rmi)

Company Links & Contacts

Company Name: Smart Programming (+91 62838-30308)

Address : Chandigarh & Mohali (Punjab), India

Websites: https://www.smartprogramming.in/

https://courses.smartprogramming.in

Android App: https://play.google.com/store/apps/details?id=com.smartprogramming

YouTube Channel: https://www.youtube.com/c/SmartProgramming

Smart Programming : YouTube Channel

An investment in Knowledge pays the best interest….

Smart Programming
We Educate - We Develop

☎ +91 62838-30308
Call us to Learn
Latest Technologies

City : Mohali (Punjab),
& Chandigarh
(India)

WEBSITE : http://www.smartprogramming.in

BUY COURSES ON : https://courses.smartprogramming.in

YOUTUBE CHANNEL : Smart Programming (https://www.youtube.com/c/SmartProgramming)

ANDROID APP : Smart Programming
(https://play.google.com/store/apps/details?id=com.smartprogramming)

https://www.facebook.com/smartprogramming.india

https://www.instagram.com/smart_programming

Java
android
python
Data Science
Machine Learning
Artificial Intelligence
M E A N
Full Stack
BIG DATA
hadoop

Encapsulation in Java

=> Encapsulation :-

-> Encapsulation is the process by which variables and methods are wrapped or binded into a single unit

-> Encapsulation is technically hiding the data from other classes and these data can be accessed only through the member functions of its own class

-> Real world example : capsule, mobile

-> Technically every java class is an example of encapsulation

-> Main example of encapsulation in java is "JavaBean" classes

-> Encapsulation is achieved by declaring variables as private and public getter and setter methods

-> For example :

```
class Employee
{
        private int salary;

        public void setSalary(int salary)
        {
                this.salary=salary;
        }
        public int getSalary()
        {
                return salary;
        }
}
```

-> In IDE's (eclipse and netbeans) getter and setter methods can be generated directly so its very easy to generate encapsulation in IDE's

-> Encapsulation = data hiding + abstraction

-> Advantages of encapsulation :-

1. Data Hiding
2. Increase flexibility
3. Reusability
4. Testing code is easy

=> JavaBean Class :-

-> JavaBean class is used to encapsulate the data into single object

-> Fules for JavaBean class :-

1. Must implement Serializable interface
2. Class must contain public no-argument constructor
3. Class must have all private variables

4. Class must contain public getter and setter methods

=> Tightly Encapsulated Class :-

-> A class is tightly encapsulated class if and only if it have all the variables as private

---

=> Interview Questions :-

1. What is difference between Abstraction & Encapsulation

-> Abstraction : It hides the implementation (details)

Encapsulation : It hides the data (information)

-> Abstraction : It is achieved by "abstract class" and "interface"

Encapsulation : It is achieved by using "JavaBean" class

-> Abstraction : In case of abstraction we have to use "abstract" keyword

Encapsulation : In case of encapsulation we have to use access modifiers (private & public)

-> Abstraction : It resolves the issue at design level

Encapsulation : It resolves the issue at implementation level

=> Which are 6 main pillers in OOP's

1. Classes
2. Objects & Methods
3. Inheritance
4. Polymorphism
5. Abstraction
6. Encapsulation

Company Links & Contacts

Company Name: Smart Programming (+91 62838-30308)

Address : Chandigarh & Mohali (Punjab), India

Websites: https://www.smartprogramming.in/
https://courses.smartprogramming.in

Android App:
https://play.google.com/store/apps/details?id=com.smartprogramming

YouTube Channel:
https://www.youtube.com/c/SmartProgramming

Smart Programming : YouTube Channel

An investment in Knowledge pays the best interest….

Smart Programming
We Educate - We Develop

+91 62838-30308
Call us to Learn
Latest Technologies

City : Mohali (Punjab), & Chandigarh (India)

WEBSITE : http://www.smartprogramming.in

BUY COURSES ON : https://courses.smartprogramming.in

YOUTUBE CHANNEL : Smart Programming (https://www.youtube.com/c/SmartProgramming)

ANDROID APP : Smart Programming (https://play.google.com/store/apps/details?id=com.smartprogramming)

https://www.facebook.com/smartprogramming.india

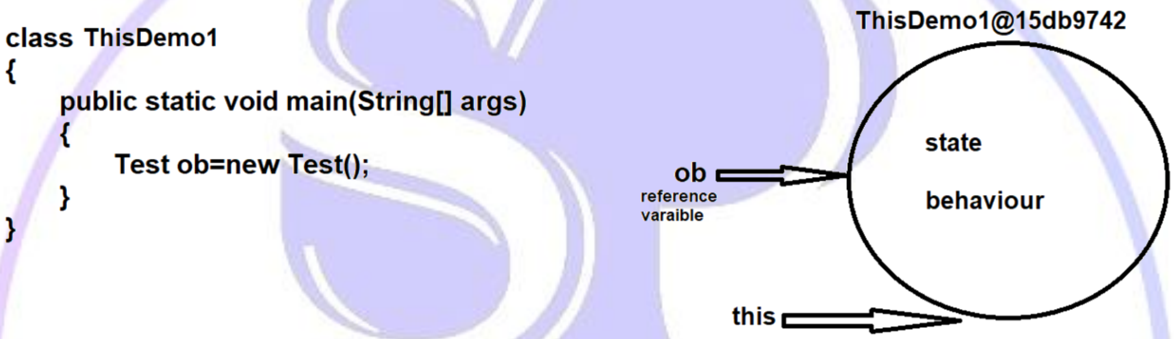
https://www.instagram.com/smart_programming

Java
android
python
Data Science
Machine Learning
Artificial Intelligence
M E A N
Full Stack
BIG DATA
hadoop

“this” keyword in Java

=> this keyword :-

-> this keyword is "reference variable" that refers to the current object



```
class ThisDemo1
{
	public static void main(String[] args)
	{
		Test ob=new Test();
	}
}
```

ThisDemo1@15db9742
state
behaviour
ob
reference
varaible
this

-> Use of this keyword :-

1. this keyword is used to refer the current class instance variable

-> this.instance_variable_name;

2. this keyword is used to invoke the current class method

-> this.methodName();

3. this keyword is used to invoke the current class constructor

-> this();

-> this(-,-,-,-);

-> this keyword must be the first statement in the constructor call

4. this keyword can be used to pass as an argument in the method

-> This case is mainly used in event handling

5. this keyword can be used to pass as an argument in the constructor

6. this keyword can be used to return current class instance

Interview Questions :-

1. Write a Program to prove that this keyword also refers to current object

2. What is constructor chaining

Company Links & Contacts

Company Name: Smart Programming (+91 62838-30308)

Address : Chandigarh & Mohali (Punjab), India

Websites: https://www.smartprogramming.in/

https://courses.smartprogramming.in

Android App:
https://play.google.com/store/apps/details?id=com.smartprogramming

YouTube Channel:
https://www.youtube.com/c/SmartProgramming

Smart Programming : YouTube Channel

An investment in Knowledge pays the best interest....

Smart Programming
We Educate - We Develop

☎ +91 62838-30308
Call us to Learn Latest Technologies
City : Mohali (Punjab), & Chandigarh (India)

WEBSITE : http://www.smartprogramming.in

BUY COURSES ON : https://courses.smartprogramming.in

YOUTUBE CHANNEL : Smart Programming (https://www.youtube.com/c/SmartProgramming)

ANDROID APP : Smart Programming (https://play.google.com/store/apps/details?id=com.smartprogramming)

https://www.facebook.com/smartprogramming.india

https://www.instagram.com/smart_programming

Java
android
python
Data Science
Machine Learning
Artificial Intelligence
M E A N
Full Stack
BIG DATA
hadoop

"super" and "final" keyword in Java

=> super keyword :

-> The super keyword is a reference variable which is used to refer immediate parent class object

-> Use of super keyword :-

1. super keyword can be used to refer the immediate parent class instance variable

2. super keyword can be used to invoke parent class method

3. super keyword is used to invoke parent class constructor

    -> If we dont provide super() in constructor then compiler provides it implicitly

    -> super() should always be the first statement in constructor

Points to remember :-

1. We cannot use this() and super() together

=> final :

-> final keyword is used to provide restrictions to the users

-> final keyword can be used with :-

1. variable (variable value cannot be changed, we cannot re-assign final variable value)
2. method (method cannot be overrided)
3. class (class cannot be inherited)

Interview Question :-

=> What is difference between this and this()

=> What is difference between super and super()

=> What is difference between final, finally and finalize

Company Links & Contacts

Company Name: Smart Programming (+91 62838-30308)

Address : Chandigarh & Mohali (Punjab), India

Websites: https://www.smartprogramming.in/

https://courses.smartprogramming.in

Android App:
https://play.google.com/store/apps/details?id=com.smartprogramming

YouTube Channel:
https://www.youtube.com/c/SmartProgramming

Smart Programming : YouTube Channel

An investment in Knowledge pays the best interest....

Smart Programming
We Educate - We Develop

+91 62838-30308
Call us to Learn
Latest Technologies

City : Mohali (Punjab),
& Chandigarh
(India)

WEBSITE : http://www.smartprogramming.in

BUY COURSES ON : https://courses.smartprogramming.in

YOUTUBE CHANNEL : Smart Programming (https://www.youtube.com/c/SmartProgramming)

ANDROID APP : Smart Programming
(https://play.google.com/store/apps/details?id=com.smartprogramming)

https://www.facebook.com/smartprogramming.india

https://www.instagram.com/smart_programming

Java
android
python
Data Science
Machine Learning
Artificial Intelligence
M E A N
Full Stack
BIG DATA
hadoop

Nested Classes in Java

=> Java Nested Class :-

-> A class which is defined inside another class or interface is known as nested class

-> Use of nested class :-

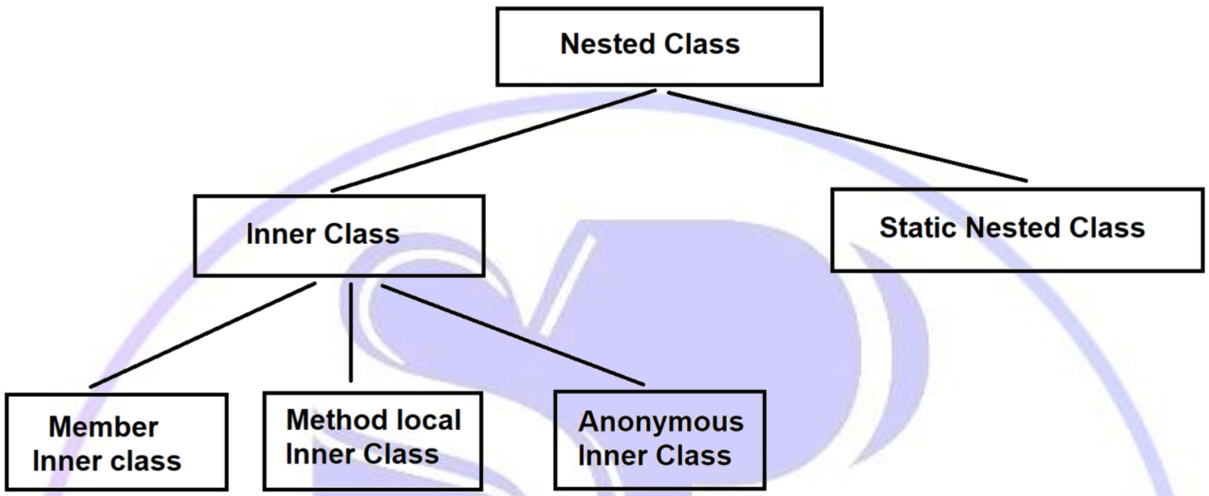
1. Nested class enable you to logically group classes that are only used in on place which will increase the use of encapsulation

2. Nested class creates more readable and maintainable code

3. Nested class help to access the private members of outer class

-> Types of Nested Class :-

1. Inner class
    - -> Member Inner Class
    - -> Method Local Inner Class
    - -> Anonymous Inner Class
2. Static Nested class


Nested Class
Inner Class
Static Nested Class
Member Inner class
Method local Inner Class
Anonymous Inner Class

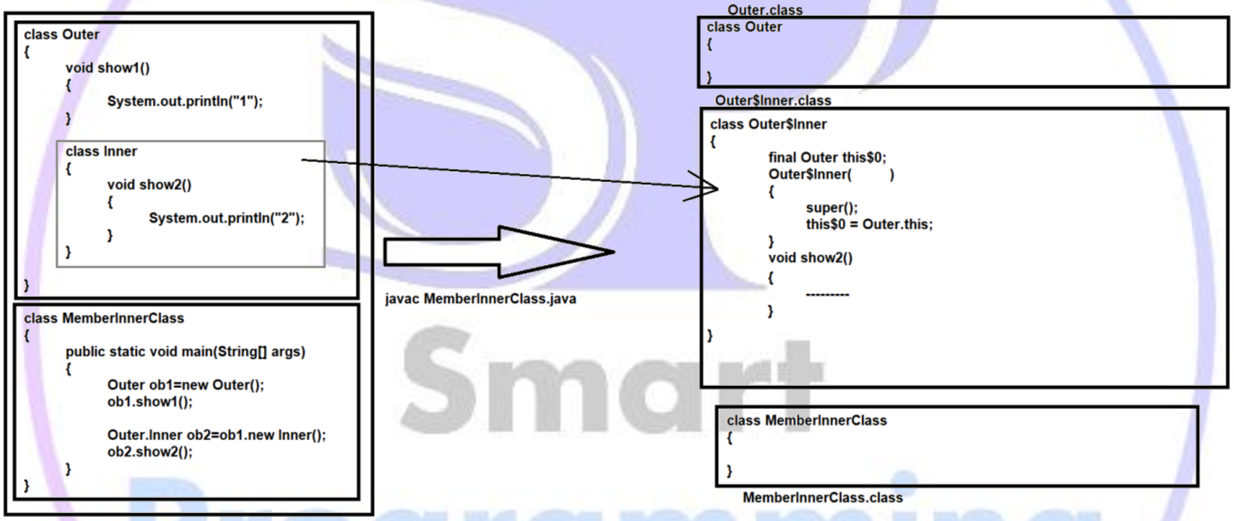
=> Member Inner Class

-> A class created within another class but outside the method is known as Member Inner Class

-> Syntax :

```
class Outer
{
    class Inner
    {
        }
    }
```

-> Program :



```
class Outer
{
        void show1()
        {
                System.out.println("1");
        }
        class Inner
        {
                void show2()
                {
                        System.out.println("2");
                }
        }
}
class MemberInnerClass
{
        public static void main(String[] args)
        {
                Outer ob1=new Outer();
                ob1.show1();

                Outer.Inner ob2=ob1.new Inner();
                ob2.show2();
        }
}
```

javac MemberInnerClass.java
Outer.class

```
class Outer
{

}
```

Outer$Inner.class

```
class Outer$Inner
{
        final Outer this$0;
        Outer$Inner(        )
        {
                super();
                this$0 = Outer.this;
        }
        void show2()
        {
                ---------
        }
}
```

```
class MemberInnerClass
{

}
```

MemberInnerClass.class

-> Points to remember :-

1. Inner class name will be generated as OuterClassName$InnerClassName

2. Automatically outer class reference is created inside inner class

=> Method Local Inner Class :-

-> When we create a class inside the method of another class, then it is known as method local inner class

-> Syntax :

```
class Outer
{
        void methodName()
        {
                class Inner
                {


                }
        }
}
```

-> Points to remember :

1. We cannot invoke inner class method from outside the method

---

=> Anonymous Inner Class

-> A class that have no name and its name is generated by the compiler automatically, that class is known as anonymous inner class

-> It is used if we want to override method of class or interface

Company Links & Contacts

Company Name: Smart Programming (+91 62838-30308)

Address : Chandigarh & Mohali (Punjab), India

Websites: https://www.smartprogramming.in/

https://courses.smartprogramming.in

Android App:
https://play.google.com/store/apps/details?id=com.smartprogramming

YouTube Channel:
https://www.youtube.com/c/SmartProgramming

Smart Programming : YouTube Channel

An investment in Knowledge pays the best interest….

Smart Programming
We Educate - We Develop

+91 62838-30308
Call us to Learn
Latest Technologies

City : Mohali (Punjab),
& Chandigarh
(India)

WEBSITE : http://www.smartprogramming.in

BUY COURSES ON : https://courses.smartprogramming.in

YOUTUBE CHANNEL : Smart Programming (https://www.youtube.com/c/SmartProgramming)

ANDROID APP : Smart Programming
(https://play.google.com/store/apps/details?id=com.smartprogramming)

https://www.facebook.com/smartprogramming.india

https://www.instagram.com/smart_programming

Java
android
python
Data Science
Machine Learning
Artificial Intelligence
M E A N
Full Stack
BIG DATA
hadoop

“Static Methods” & “Static Nested Class” in Java

=> static methods :-

-> If we use static keyword before method, then it is static method

-> Instance methods belong to an instance (object) but static methods belongs to the class

-> Syntax : static access-modifier return-type methodName()

```
{


}
```

=> Points to remember :-

-> We can only access static members within the static method

-> We cannot use this or super keyword in static methods

-> We cannot create static constructor

=> Why main method is static ?

Main method is static because JVM can call main method without creating an object.

=> Static Nested Class :-

-> If we create static nested class within another class, it is known as static nested class

-> Example :-

```
class Outer
{
        static class Inner
        {


        }
}
```

-> Outer class cannot be created as static class

-> Points to remember :-

1. Static nested class can access only static variables of outer class

Company Links & Contacts

Company Name: Smart Programming (+91 62838-30308)

Address : Chandigarh & Mohali (Punjab), India

Websites: https://www.smartprogramming.in/

https://courses.smartprogramming.in

Android App: https://play.google.com/store/apps/details?id=com.smartprogramming

YouTube Channel: https://www.youtube.com/c/SmartProgramming

Smart Programming : YouTube Channel

An investment in Knowledge pays the best interest....

Smart Programming
We Educate - We Develop

+91 62838-30308
Call us to Learn
Latest Technologies

City : Mohali (Punjab),
& Chandigarh
(India)

WEBSITE : http://www.smartprogramming.in

BUY COURSES ON : https://courses.smartprogramming.in

YOUTUBE CHANNEL : Smart Programming (https://www.youtube.com/c/SmartProgramming)

ANDROID APP : Smart Programming
(https://play.google.com/store/apps/details?id=com.smartprogramming)

https://www.facebook.com/smartprogramming.india

https://www.instagram.com/smart_programming

Java
android
python
Data Science
Machine Learning
Artificial Intelligence
M E A N
Full Stack
BIG DATA
hadoop

Pre-Defined Packages in Java

=> Packages :-

-> A package is the group of similar type of classes or interfaces or sub-packages

-> A package is a folder which contains classes or interfaces (.class) or sub-packages

-> Advantages of packages :-

1. Packages are used to categorize the classes or interfaces which makes them easy to maintain or access
2. Packages are used to achieve modularity in our project
3. Packages are used to achieve abstraction in java
4. Packages are used to achieve security in java
5. Packages can be used to remove nameing collision

6. Packages improve shareability because we can use same package with different java applications

7. Packages are used to achieve reusability

-> Types of packages :-

1. Predefined Package (Built-in Package)
2. User Defined Package

-> How to import the packages :-

1. import packagename.subpackage.*;
2. import packagename.subpackage.ClassName/InterfaceName;
3. Fully Qualified ClassName/InterfaceName

=> Predefined Package (Built-in Package) :-

-> The packages which are already provided by java are known as predefined packages

-> Important types of packages in core java :-

1. java.lang
2. java.util
3. java.io
4. java.awt
5. javax.swing
6. java.sql
7. java.applet
8. java.net
9. java.rmi

=> java.lang :-

-> java.lang package contains the basic classes used in java programs

-> Classes/Iterfaces/subpackages are :-

= Object, Class, reflect (sub-package - Method, Constructor etc) etc

= System

= Boolean, Character, Byte, Short, Integer, Long, Float, Double

= Exception, NullPointerException, ArithmeticException, ArrayIndexOutOfBoundException etc.

= String, StringBuffer, StringBuilder

= Thread, Runnable

= more classes and interfaces like Comparable, Cloneable...

-> How to import this package :-

import java.lang.*;

-> java.lang package is the only package which is by default imported to every java class file

---

=> java.util :-

-> java.util package contains the utility classes and data-structure related classes

-> Classes/Iterfaces/subpackages are :-

= Scanner, Calender, Date, Locale etc

= Collection, List, Set, Queue, ArrayList, LinkedList, Vector, TreeSet etc

= regex (subpackage) Pattern, Matcher etc

-> How to import this package :-

import java.util.*;

=> java.io :-

-> java.io package contains the input and output operations related classes or interfaces

-> Classes/Iterfaces/subpackages :-

= InputStream, OutputStream, FileInputStream, FileOutputStream etc

= Reader, Writer etc

-> How to import this package :-

`import java.io.*;`

=> java.awt :-

-> java.awt package contains GUI based classes and interfaces

-> Classes/Iterfaces/subpackages :-

= Window, Frame, Label, Button, TextField, CheckBox, etc

-> How to import this package :-

`import java.awt.*;`

=> javax.swing :-

-> javax.swing contains GUI based classes and interfaces

-> Classes/Iterfaces/subpackages :-

= JWindow, JFrame, JTextfield, JButton etc

-> How to import this package :-

`import javax.swing.*;`

=> java.sql :-

-> java.sql package contains database related classes or interfaces

-> Classes/Iterfaces/subpackages :-

= Driver, DriverManager, Statement, PreparedStatement, ResultSet, CallableStatement, ResultSetMetaData etc

-> How to import this package :-

`import java.sql.*;`

=> java.applet :-

-> java.applet package provides GUI based classes and interfaces

-> Classes/Iterfaces/subpackages :-

= Frame, Button, Graphics, Textfield etc

-> How to import this package :-

import java.applet.*;

=> java.net :-

-> java.net package contains the classes and interface which are used to create Distributed Applications.

-> Main task we achieve from this package is Socket Programming

-> Classes/Iterfaces/subpackages :-

= Socket, ServerSocket, URLConnection, URL, InetAddress etc

-> How to import this package :-

import java.net.*;

=> java.rmi :-

-> RMI : Remote Method Invocation

-> java.rmi package contains classes or interfaces which are used to prepare Distributed Applications on the basis of RMI

-> Classes/Iterfaces/subpackages :-

= Remote, RemoteException, Naming, UnicastRemoteObject etc

-> How to import this package :-

`import java.rmi.*;`

Company Links & Contacts

Company Name: Smart Programming (+91 62838-30308)

Address : Chandigarh & Mohali (Punjab), India

Websites: https://www.smartprogramming.in/

https://courses.smartprogramming.in

Android App: https://play.google.com/store/apps/details?id=com.smartprogramming

YouTube Channel: https://www.youtube.com/c/SmartProgramming

Smart Programming : YouTube Channel

An investment in Knowledge pays the best interest....

Smart Programming
We Educate - We Develop

+91 62838-30308
Call us to Learn
Latest Technologies

City : Mohali (Punjab),
& Chandigarh
(India)

WEBSITE : http://www.smartprogramming.in

BUY COURSES ON : https://courses.smartprogramming.in

YOUTUBE CHANNEL : Smart Programming (https://www.youtube.com/c/SmartProgramming)

ANDROID APP : Smart Programming
(https://play.google.com/store/apps/details?id=com.smartprogramming)

https://www.facebook.com/smartprogramming.india

https://www.instagram.com/smart_programming

Java
android
python
Data Science
Machine Learning
Artificial Intelligence
M E A N
Full Stack
BIG DATA
hadoop

User-Defined Packages in Java

=> User Defined Package :-

-> The package created by the user or developer according to the project requirements is known as User Defined Package

-> Syntax :

package package_name;

package package_name.sub_package_name;

-> Naming Convention for Package :-

1. Same 4 rules as an identifiers

2. -> company domain in reverse order

   -> client name

   -> project name

   -> module name

   => For example :-

   www.smartprogramming.in (my company domain)

ICICI (client)

Loan (Project name)

Home Loan, Car Loan, Jewellery Loan etc - Sub-package

-> in.smartprogramming.icici.loan.home_loan

-> in.smartprogramming.icici.loan.car_loan

-> Program :-

-> Step 1 :-

```
package pkg;
class PackageDemo
{
	public static void main(String[] args)
	{
		System.out.println("hello");
	}
```

}

-> Step 2:

1. javac -d . PackageDemo.java (d represents the directory and . represents the current directory)

2. javac -d C:\aaa PackageDemo.java

-> Step 3:

1. java pkg.PackageDemo

2. set classpath=C:\aaa

java pkg.PackageDemo

-> Points to remember :-

1. Package declaration must be the first statement in java program

2. There can be only one package statement

=> How to import packages :-

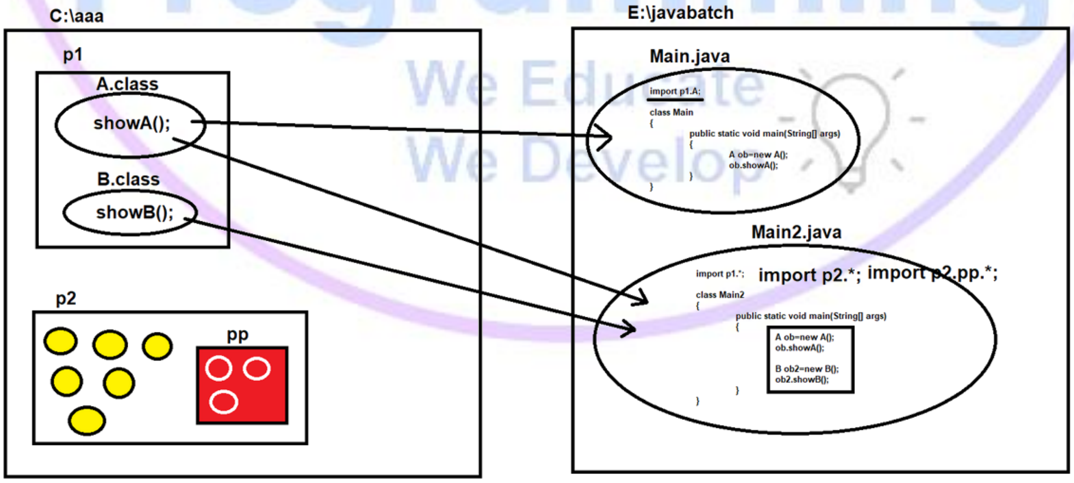
-> There are 3 ways to import packages in java :-

1. by using import package_name.ClassName;

   -> It will import only particular class file from the provided package name

2. by using import package_name.*;

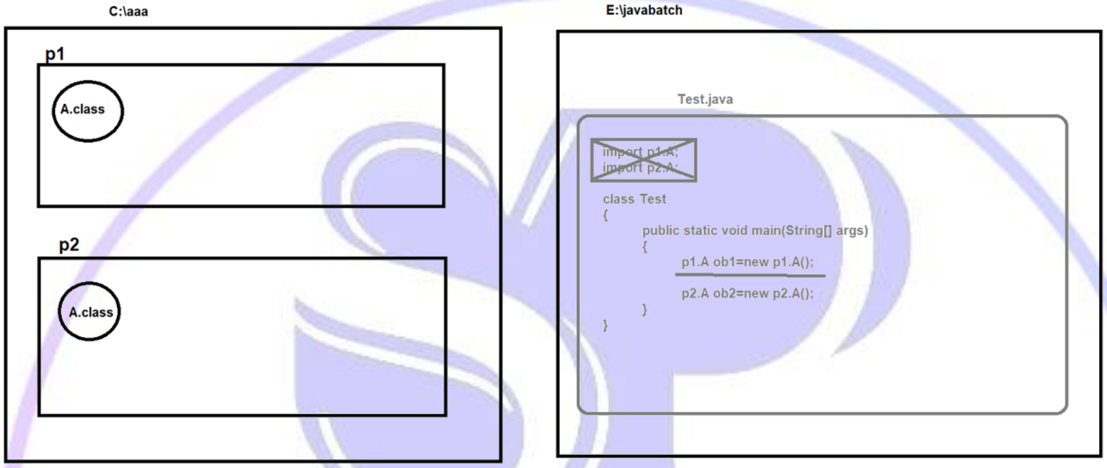
   -> It will import all the classes from provided package name


C:\aaa
p1
A.class
showA();
B.class
showB();
p2
pp
E:\javabatch
Main.java
Main2.java

```
import p1.A;
class Main
{
        public static void main(String[] args)
        {
                A ob=new A();
                ob.showA();
        }
}
```

```
import p1.*;   import p2.*; import p2.pp.*;
class Main2
{
        public static void main(String[] args)
        {
                A ob=new A();
                ob.showA();

                B ob2=new B();
                ob2.showB();
        }
}
```

3. by using fully qualified ClassName;



```
import p1.A;
import p2.A;

class Test
{
        public static void main(String[] args)
        {
                p1.A ob1=new p1.A();

                p2.A ob2=new p2.A();
        }
}
```

-> Points to remember :-

1. We can use any number of import statement

2. import statement will come immediate after the package statement

3. import statement does not load the class in the memory, it only provides the accessability to use the classes

4. If we use import statement, then only class will be imported but sub-packages will not import

---

Interview Question :-

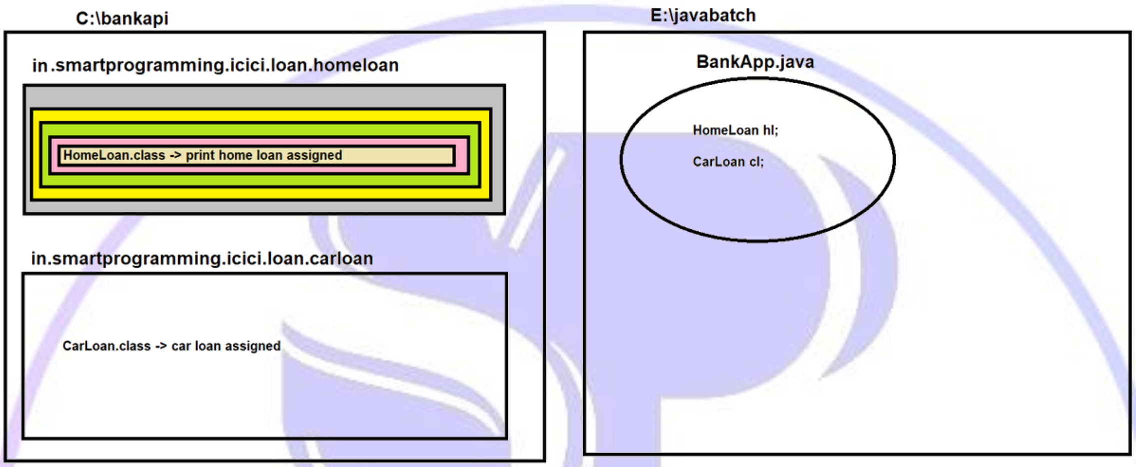
1. What is difference between set path and set classpath ?

-> By using set path=C:\Program Files\Java\jdk1.8.0_45\bin, System will understand java commands with the help of path variable

-> By using set classpath=----, it will set the classpath where the .class file is saved

---

Task :-


C:\bankapi
in.smartprogramming.icici.loan.homeloan
HomeLoan.class -> print home loan assigned
in.smartprogramming.icici.loan.carloan
CarLoan.class -> car loan assigned
E:\javabatch
BankApp.java
HomeLoan hl;
CarLoan cl;

Company Links & Contacts

Company Name: Smart Programming (+91 62838-30308)

Address : Chandigarh & Mohali (Punjab), India

Websites: https://www.smartprogramming.in/

https://courses.smartprogramming.in

Android App: https://play.google.com/store/apps/details?id=com.smartprogramming

YouTube Channel: https://www.youtube.com/c/SmartProgramming

Smart Programming : YouTube Channel

An investment in Knowledge pays the best interest....

Smart Programming
We Educate - We Develop

☎ +91 62838-30308
Call us to Learn
Latest Technologies

City : Mohali (Punjab),
& Chandigarh
(India)

WEBSITE : http://www.smartprogramming.in

BUY COURSES ON : https://courses.smartprogramming.in

YOUTUBE CHANNEL : Smart Programming (https://www.youtube.com/c/SmartProgramming)

ANDROID APP : Smart Programming
(https://play.google.com/store/apps/details?id=com.smartprogramming)

https://www.facebook.com/smartprogramming.india

https://www.instagram.com/smart_programming

Java
android
python
Data Science
Machine Learning
Artificial Intelligence
M E A N
Full Stack
BIG DATA
hadoop

Jar File & MANIFEST.MF File in Java

=> Jar Files :-

-> Full form of JAR is Java Archive

-> Jar file is used to aggregate .class files, metadata, packages, images, audio files etc into one file so that we can distribute it on java platform

-> Jar file is the compressed form of all the resources (.class, packages, metadata, images, audio etc)

-> How to create jar file :-

= jar -cf JarFileName.jar file_names (cf is create file - cvf create verbos file)

= jar -cf JarFileName.jar *

-> How to view jar file content :-

= jar -tf JarFileName.jar (tf is table format or tableview in file)

-> How to extract jar files :-

= jar -xf JarFileName.jar (xf is extract file)

-> How to update jar file :-

= jar -uf JarFileName.jar

-> How to run jar file :-

= java -jar JarFileName.jar

=> MANIFEST.MF File :-

-> Manifest is the file which is present in every jar file which represents the features of jar file

-> Whenever we create jar file, MANIFEST.MF file is created automatically in META-INF folder

-> The default Manifest file contains only 2 enteries in the form of name-value pair but we can insert more values in MANIFEST.MF file

-> How to insert data in MANIFEST.MF file :-

1. Create any txt file with any name and provide values in it using name-value pair
2. Then provide the command in CMD :-

```
jar -cfm JarFileName.jar TextFileName.txt *
```

-> Points to remember :-

1. We have to provide the space after name and colon
2. If we provide any .class file name in text file then we should not include .class at the end of ClassName
3. We should always add newline at the end of text file

=> Executable Jar File :-

-> If any jar file contains the main class (the class which contains the main method) is known as executable jar file

-> To create executable jar file, we have to provide Main-Class name value pair in manifest file

-> How to run executable jar file form CMD :-

```
java -jar JarFileName.jar
```

=> Package Sealing :-

-> We can seal a package in java to ensure that no further classes can add themselves to it.

-> For example we can use package sealing to ensure version consistency among the classes in our software

-> How to achieve package sealing :-

We have to add name-value pair in MANIFEST file i.e Sealed: true

---

=> Batch File :-

-> A batch file is a script file in windows or DOS

-> A batch file contains a series of commands

-> Extention of batch file is .bat

Company Links & Contacts

Company Name: Smart Programming (+91 62838-30308)

Address : Chandigarh & Mohali (Punjab), India

Websites: https://www.smartprogramming.in/

https://courses.smartprogramming.in

Android App:
https://play.google.com/store/apps/details?id=com.smartprogramming

YouTube Channel:
https://www.youtube.com/c/SmartProgramming

Smart Programming : YouTube Channel

An investment in Knowledge pays the best interest....

Smart Programming
We Educate - We Develop

☎ +91 62838-30308
Call us to Learn
Latest Technologies

City : Mohali (Punjab),
& Chandigarh
(India)

WEBSITE : http://www.smartprogramming.in

BUY COURSES ON : https://courses.smartprogramming.in

YOUTUBE CHANNEL : Smart Programming (https://www.youtube.com/c/SmartProgramming)

ANDROID APP : Smart Programming
(https://play.google.com/store/apps/details?id=com.smartprogramming)

https://www.facebook.com/smartprogramming.india

https://www.instagram.com/smart_programming

Java
android
python
Data Science
Machine Learning
Artificial Intelligence
M E A N
Full Stack
BIG DATA
hadoop

AWT & Swing in Java

GUI Part :-

-> GUI is Graphical User Interface

-> Java Provides following GUI Part :-

1. AWT (Abstract Window Toolkit)
2. Swing
3. Applet
4. javafx

=> AWT

-> AWT is an API used to develop Desktop Based or Window Based Application

-> AWT has a lot of disadvantages and due to this, Swing API was created on the basis of AWT

=> Swing :-

-> Swing is also a GUI part which is used to create Desktop Based Applications

---

=> Difference between AWT & Swing :-

1. AWT :- AWT provides heavy-weight components

Swing : Swing provides light-weight components

2. AWT : AWT is old framework

Swing : Swing is new framework

3. AWT : To use AWT we have to import java.awt package

Swing : To use Swing we have to import java.swing package

4. AWT : AWT provides less number of components

Swing : Swing provides more number of components

5. AWT : AWT is not portable

Swing : Swing is portable

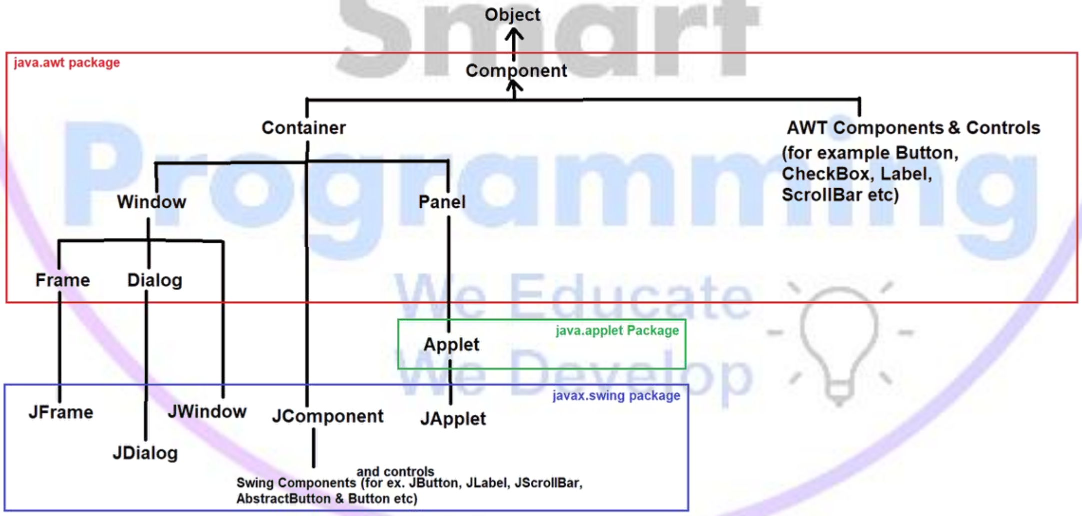
6. AWT : AWT requires more code for control functionalities

Swing : Swing requires less code for control functionalities

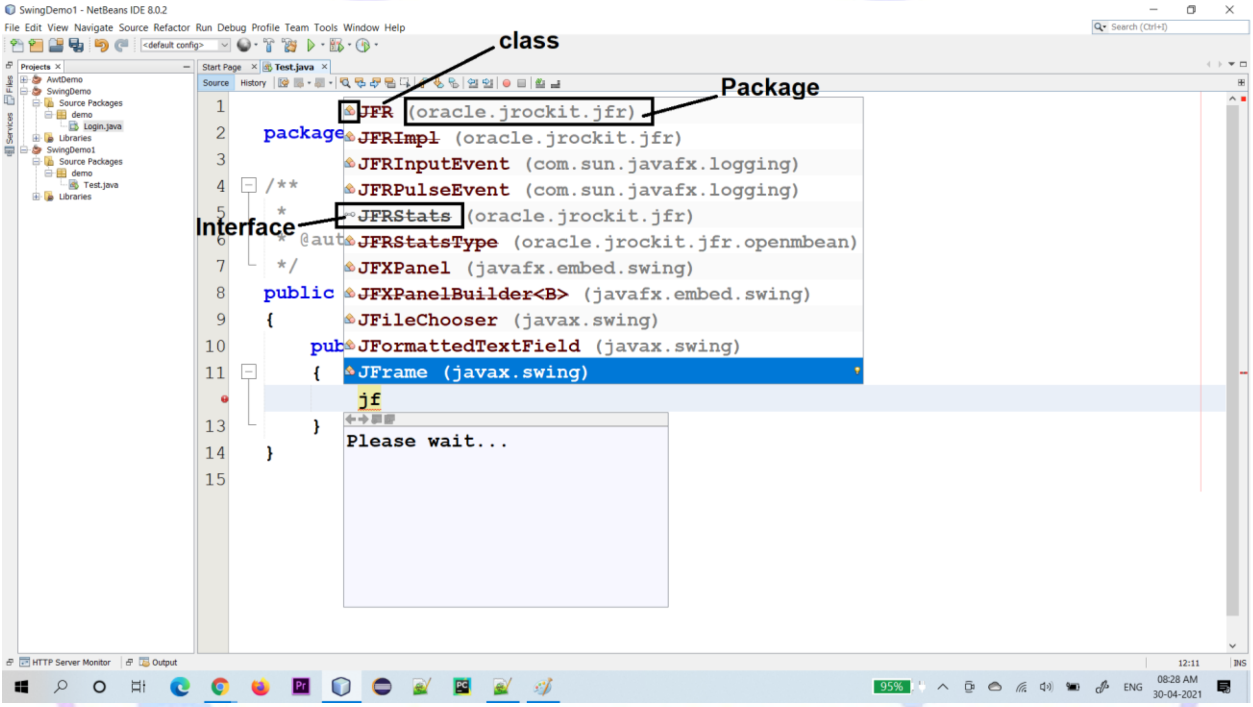
7. AWT : AWT does not support look and feel pluggable

Swing : Swing supports look and feel pluggable

=> Heirarchy of AWT & Swing :-


Object
java.awt package
Component
Container
AWT Components & Controls
(for example Button, CheckBox, Label, ScrollBar etc)
Window
Panel
Frame
Dialog
java.applet Package
Applet
javax.swing package
JFrame
JWindow
JComponent
JApplet
JDialog
Swing Components (for ex. JButton, JLabel, JScrollBar, AbstractButton & Button etc) and controls

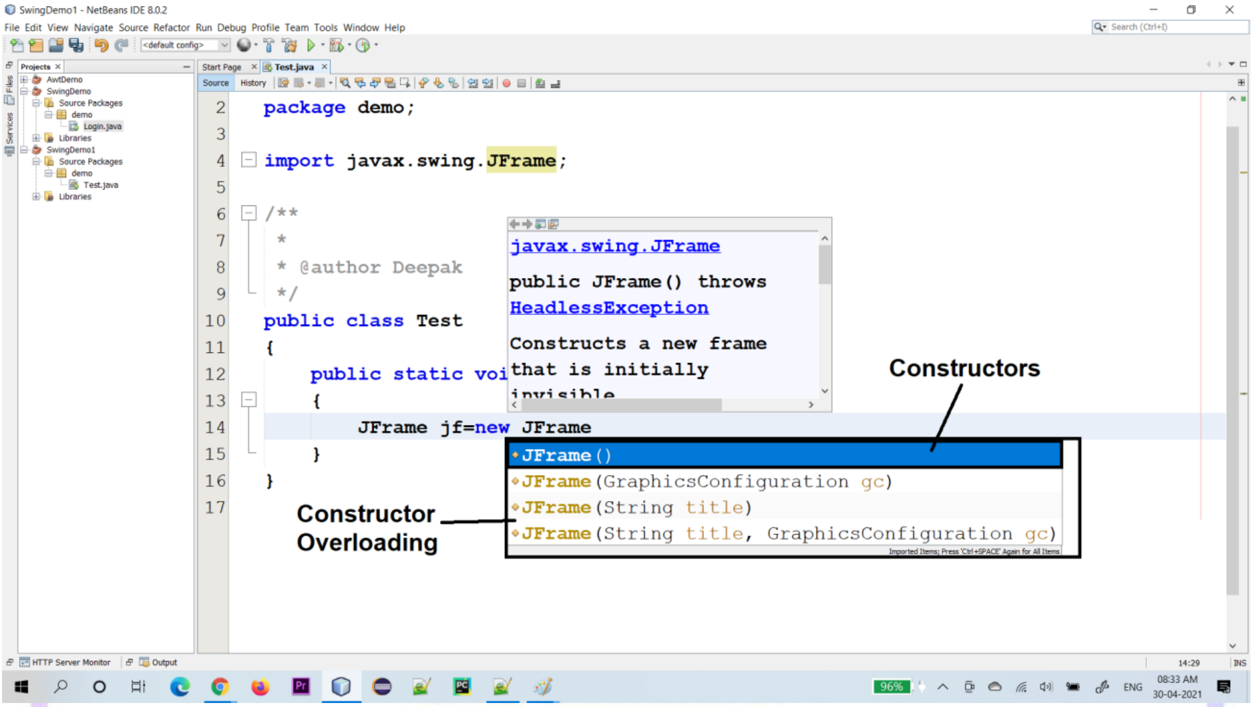
Netbeans Basic Thing You Should Know :-


SwingDemo1 - NetBeans IDE 8.0.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
class
Package
Interface
Projects
Start Page
Test.java
Source
History

```
package
/**
 *
 * @aut
 */
public
{
    pub
    {
        jf
    }
}
```

```
JFR  (oracle.jrockit.jfr)
JFRImpl  (oracle.jrockit.jfr)
JFRInputEvent  (com.sun.javafx.logging)
JFRPulseEvent  (com.sun.javafx.logging)
JFRStats  (oracle.jrockit.jfr)
JFRStatsType  (oracle.jrockit.jfr.openmbean)
JFXPanel  (javafx.embed.swing)
JFXPanelBuilder<B>  (javafx.embed.swing)
JFileChooser  (javax.swing)
JFormattedTextField  (javax.swing)
JFrame  (javax.swing)
```

```
Please wait...
```



```
package demo;

import javax.swing.JFrame;

/**
 *
 * @author Deepak
 */
public class Test
{
    public static voi
    {
        JFrame jf=new JFrame
    }
}
```

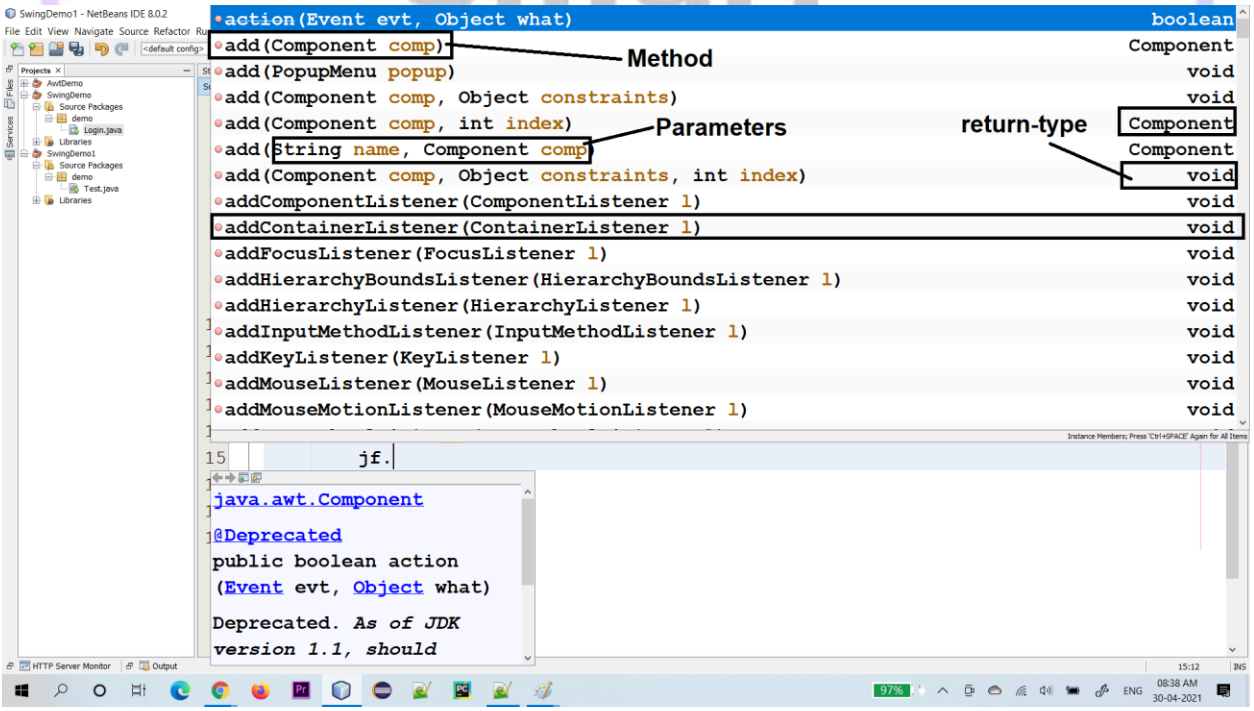
javax.swing.JFrame

public JFrame() throws HeadlessException

Constructs a new frame that is initially invisible

```
JFrame()
JFrame(GraphicsConfiguration gc)
JFrame(String title)
JFrame(String title, GraphicsConfiguration gc)
```

Smart



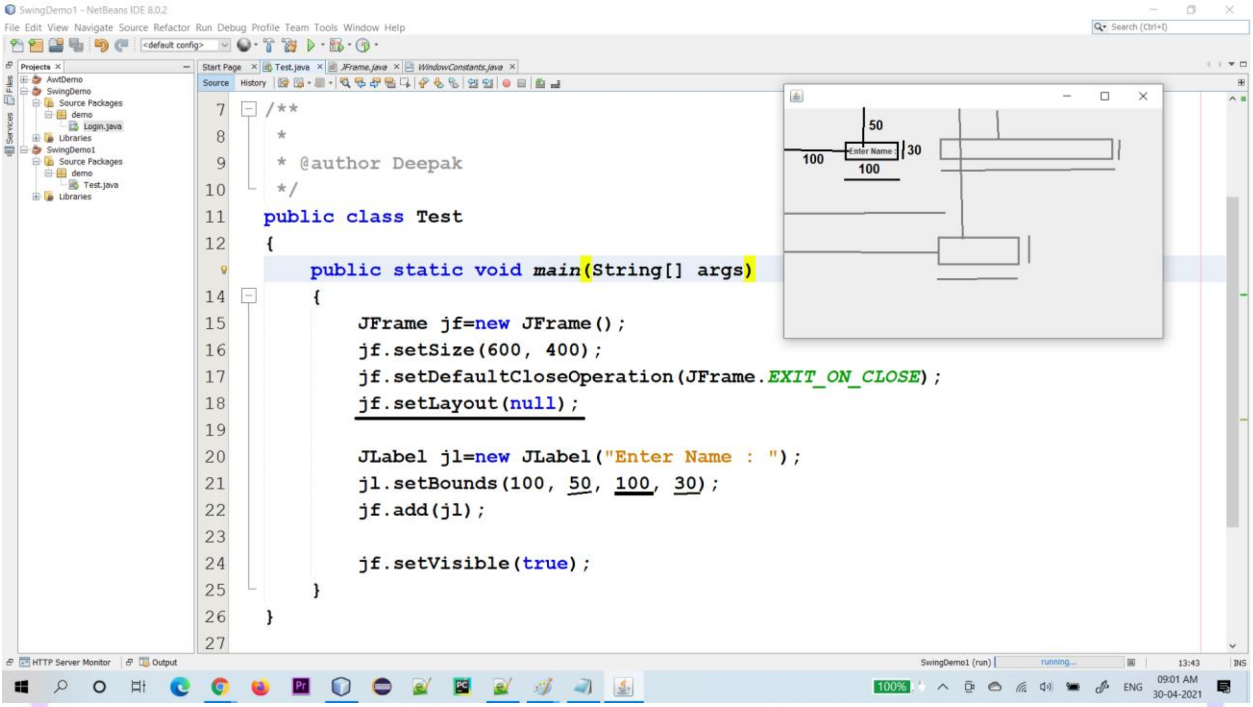
```
action(Event evt, Object what)                              boolean
add(Component comp)                                         Component
add(PopupMenu popup)                                        void
add(Component comp, Object constraints)                     void
add(Component comp, int index)                              Component
add(String name, Component comp)                            Component
add(Component comp, Object constraints, int index)          void
addComponentListener(ComponentListener l)                   void
addContainerListener(ContainerListener l)                   void
addFocusListener(FocusListener l)                           void
addHierarchyBoundsListener(HierarchyBoundsListener l)       void
addHierarchyListener(HierarchyListener l)                   void
addInputMethodListener(InputMethodListener l)               void
addKeyListener(KeyListener l)                               void
addMouseListener(MouseListener l)                           void
addMouseMotionListener(MouseMotionListener l)               void
```

```
        jf.
```

java.awt.Component

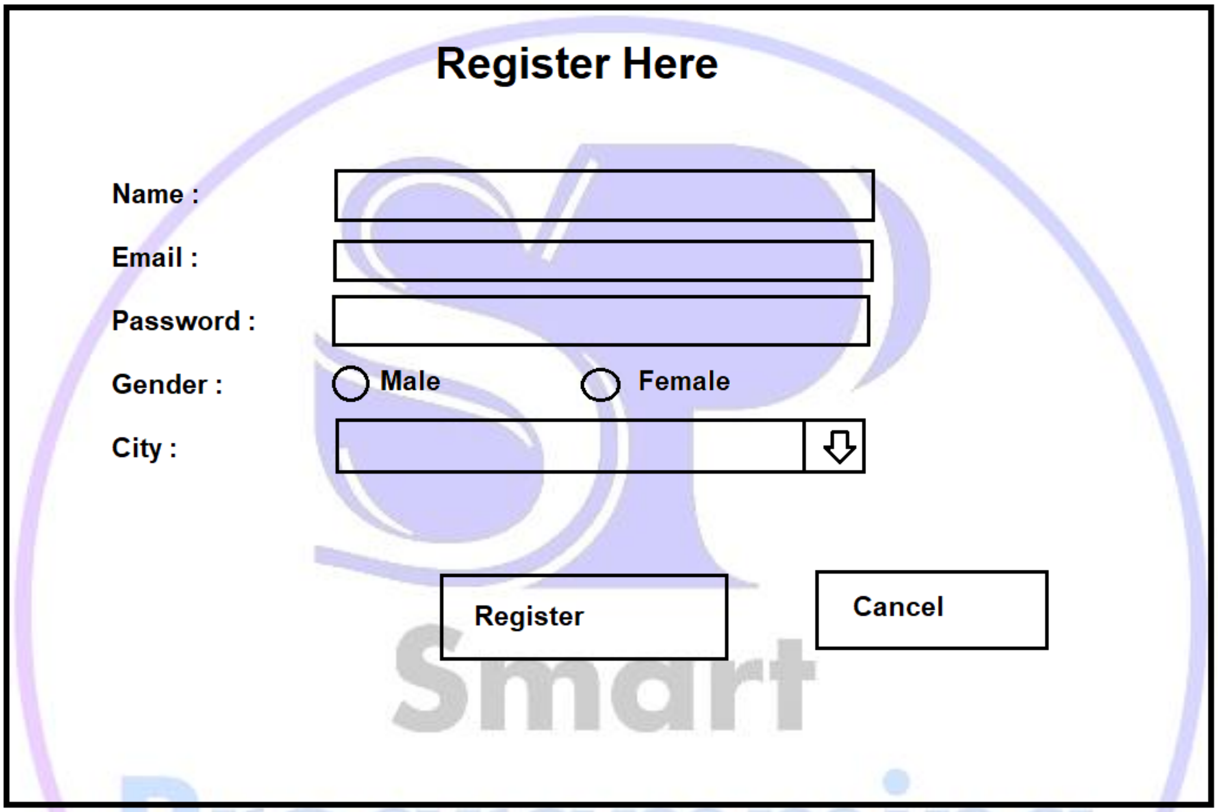
@Deprecated
public boolean action (Event evt, Object what)

Deprecated. *As of JDK version 1.1, should*


SwingDemo1 - NetBeans IDE 8.0.2
Start Page
Test.java
JFrame.java
WindowConstants.java
Source
History
50
100
Enter Name
30
100

```
/**
 *
 * @author Deepak
 */
public class Test
{
    public static void main(String[] args)
    {
        JFrame jf=new JFrame();
        jf.setSize(600, 400);
        jf.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        jf.setLayout(null);

        JLabel jl=new JLabel("Enter Name : ");
        jl.setBounds(100, 50, 100, 30);
        jf.add(jl);

        jf.setVisible(true);
    }
}
```

Task :-


Register Here
Name :
Email :
Password :
Gender :
Male
Female
City :
Register
Cancel

Company Links & Contacts

Company Name: Smart Programming (+91 62838-30308)

Address : Chandigarh & Mohali (Punjab), India

Websites: https://www.smartprogramming.in/

https://courses.smartprogramming.in

Android App: https://play.google.com/store/apps/details?id=com.smartprogramming

YouTube Channel: https://www.youtube.com/c/SmartProgramming

Smart Programming : YouTube Channel

An investment in Knowledge pays the best interest....

Smart Programming
We Educate - We Develop

+91 62838-30308
Call us to Learn
Latest Technologies

City : Mohali (Punjab),
& Chandigarh
(India)

WEBSITE : http://www.smartprogramming.in

BUY COURSES ON : https://courses.smartprogramming.in

YOUTUBE CHANNEL : Smart Programming (https://www.youtube.com/c/SmartProgramming)

ANDROID APP : Smart Programming
(https://play.google.com/store/apps/details?id=com.smartprogramming)

https://www.facebook.com/smartprogramming.india

https://www.instagram.com/smart_programming

Java
android
python
Data Science
Machine Learning
Artificial Intelligence
M E A N
Full Stack
BIG DATA
hadoop

Events in AWT

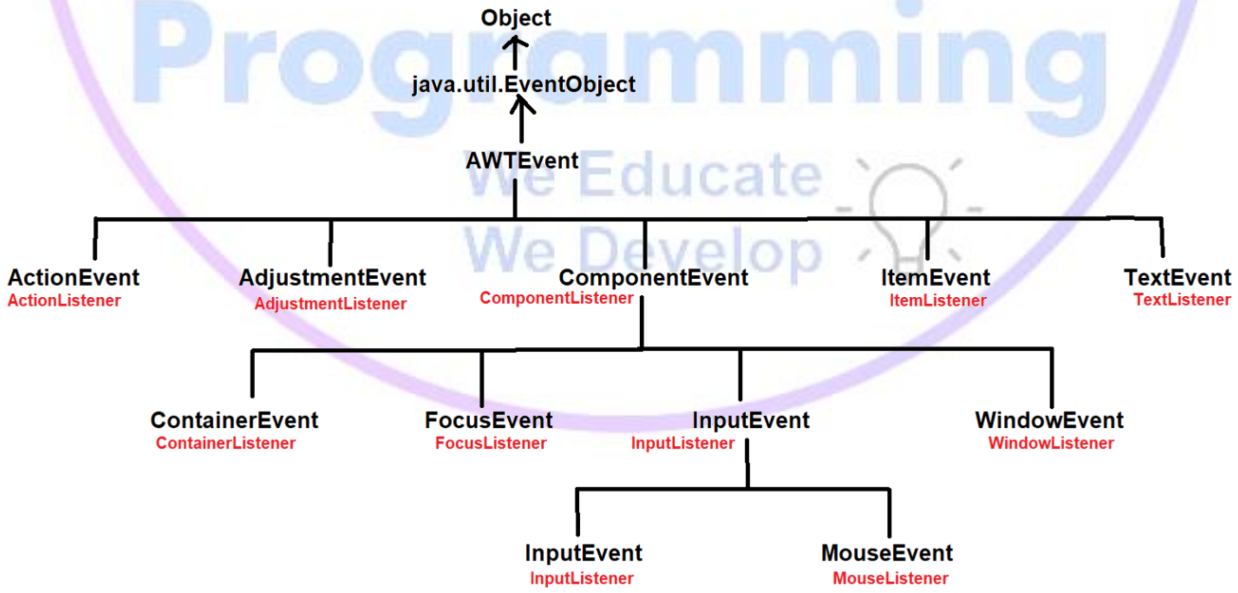
=> Event Handling :-

-> Changing the state of an object is known as event.

-> For example clicking on button, focus on textfield, mouse event on any component etc

-> To perform any event java provides an package i.e. java.awt.event

-> Hierarchy of Event Handling in AWT :-


Object
java.util.EventObject
AWTEvent
ActionEvent
ActionListener
AdjustmentEvent
AdjustmentListener
ComponentEvent
ComponentListener
ItemEvent
ItemListener
TextEvent
TextListener
ContainerEvent
ContainerListener
FocusEvent
FocusListener
InputEvent
InputListener
WindowEvent
WindowListener
InputEvent
InputListener
MouseEvent
MouseListener

-> NOTE : Whenever we have to perform any event on component then we have to register that component for event and for this java provide registration methods

Company Links & Contacts

Company Name: Smart Programming (+91 62838-30308)

Address : Chandigarh & Mohali (Punjab), India

Websites: https://www.smartprogramming.in/
https://courses.smartprogramming.in

Android App:
https://play.google.com/store/apps/details?id=com.smartprogramming

YouTube Channel:
https://www.youtube.com/c/SmartProgramming

Smart Programming : YouTube Channel

An investment in Knowledge pays the best interest….

Smart Programming
We Educate - We Develop

☎ +91 62838-30308
Call us to Learn
Latest Technologies

City : Mohali (Punjab),
& Chandigarh
(India)

WEBSITE : http://www.smartprogramming.in

BUY COURSES ON : https://courses.smartprogramming.in

YOUTUBE CHANNEL : Smart Programming (https://www.youtube.com/c/SmartProgramming)

ANDROID APP : Smart Programming
(https://play.google.com/store/apps/details?id=com.smartprogramming)

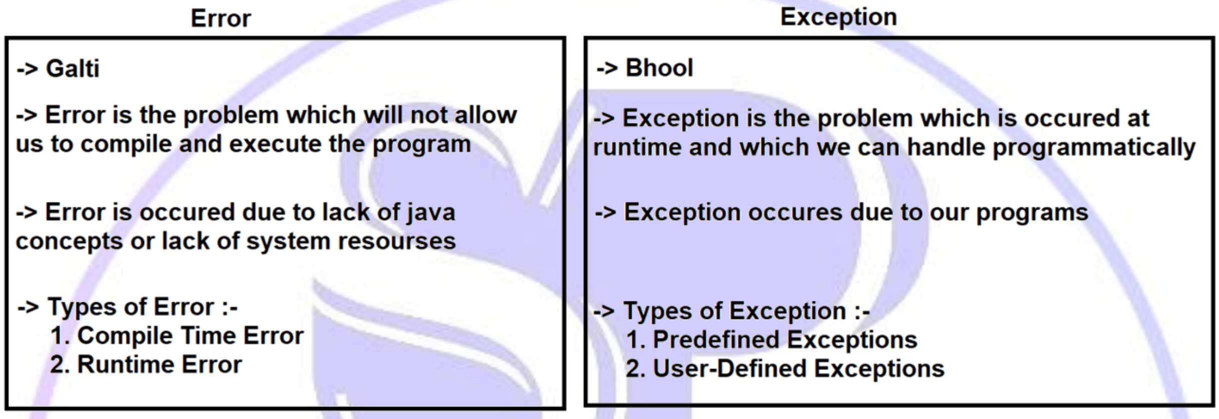
https://www.facebook.com/smartprogramming.india

https://www.instagram.com/smart_programming

Java
android
python
Data Science
Machine Learning
Artificial Intelligence
M E A N
Full Stack
BIG DATA
hadoop

Error & Exception Introduction in Java

=> What is difference between Error & Exception :-



Error	Exception
-> Galti	-> Bhool
-> Error is the problem which will not allow us to compile and execute the program	-> Exception is the problem which is occured at runtime and which we can handle programmatically
-> Error is occured due to lack of java concepts or lack of system resourses	-> Exception occures due to our programs
-> Types of Error :- 1. Compile Time Error 2. Runtime Error	-> Types of Exception :- 1. Predefined Exceptions 2. User-Defined Exceptions

=> Types of Errors :-

- ➔ Compile Time Error
    - ○ Lexical Error (if we have any mistake in java keywords for example static - statc, void - viod)
    - ○ Syntax Error (if we have any mistake in java predefined syntax for example variable declaration, method declaration, method calling etc)

  - ○ Semantic Error (it is the meaning-less statement)
  - ○ Etc

➔ Runtime Error

1. Runtime errors cannot be handled programmatically

2. Runtime errors occurs because of JVM (memory error)

=> Exception :-

➔ Exception is any unwanted event which disturb the normal flow of the program

➔ Types of Exceptions :-

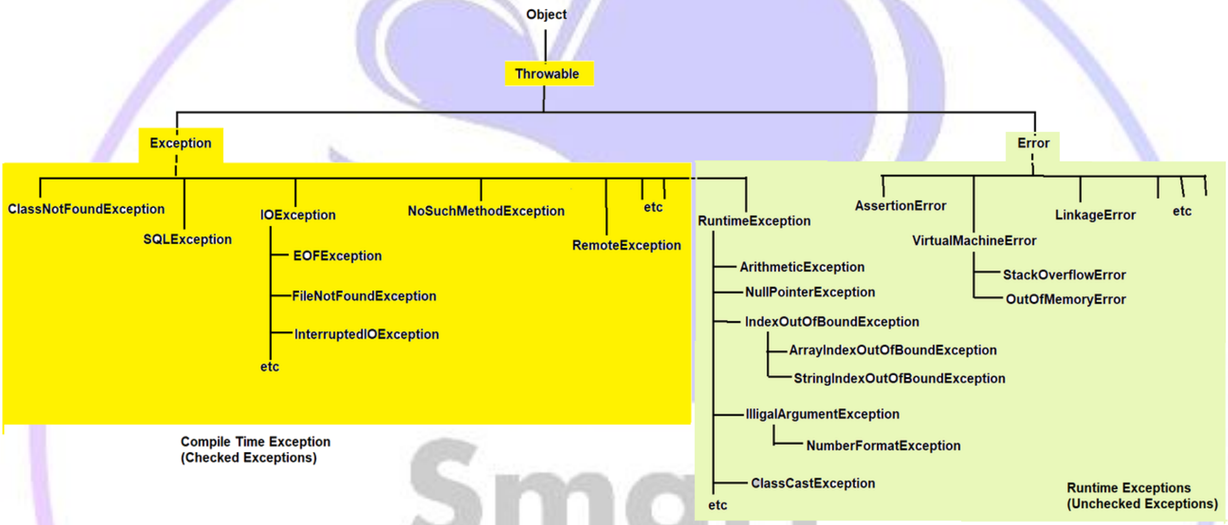
= Predefined Exceptions

= User-Defined Exceptions

(above both can be of 2 types - Checked Exceptions & Unchecked Exceptions)

Smart
Programming
We Educate
We Develop

=> Hierarchy of Error & Exception class :-


Object
Throwable
Exception
Error
ClassNotFoundException
SQLException
IOException
EOFException
FileNotFoundException
InterruptedIOException
etc
NoSuchMethodException
etc
RemoteException
Compile Time Exception
(Checked Exceptions)
RuntimeException
ArithmeticException
NullPointerException
IndexOutOfBoundException
ArrayIndexOutOfBoundException
StringIndexOutOfBoundException
IlligalArgumentException
NumberFormatException
ClassCastException
etc
AssertionError
VirtualMachineError
StackOverflowError
OutOfMemoryError
LinkageError
etc
Runtime Exceptions
(Unchecked Exceptions)

Company Links & Contacts

Company Name: Smart Programming (+91 62838-30308)

Address : Chandigarh & Mohali (Punjab), India

Websites: https://www.smartprogramming.in/

https://courses.smartprogramming.in

Android App: https://play.google.com/store/apps/details?id=com.smartprogramming

YouTube Channel: https://www.youtube.com/c/SmartProgramming

Smart Programming : YouTube Channel

An investment in Knowledge pays the best interest....

Smart Programming
We Educate - We Develop

☎ +91 62838-30308
Call us to Learn Latest Technologies

City : Mohali (Punjab), & Chandigarh (India)

WEBSITE : http://www.smartprogramming.in

BUY COURSES ON : https://courses.smartprogramming.in

YOUTUBE CHANNEL : Smart Programming (https://www.youtube.com/c/SmartProgramming)

ANDROID APP : Smart Programming (https://play.google.com/store/apps/details?id=com.smartprogramming)

https://www.facebook.com/smartprogramming.india

https://www.instagram.com/smart_programming

Java
android
python
Data Science
Machine Learning
Artificial Intelligence
M E A N
Full Stack
BIG DATA
hadoop

Checked & Unchecked Exception in Java

=> What is Checked and Unchecked Exception :-

➔ Checked exception is the exception which can be checked by the compiler

- Unchecked exception is the exception which compiler cannot check

➔ If we do not report checked exception, our program will not compile

- In case of Unchecked exception, if we do not report the exception then our program can compile

➔ In case of checked exception we have to use throws or try-catch keyword

- In case of unchecked exception its not compulsory to use throws or try-catch keyword

➔ Parent class of checked exceptions are "Exception" class

Parent class of unchecked exceptions are "RuntimeException" class

---

=> NOTE : Exceptions are always occured at runtime, no exception is occured at compile time

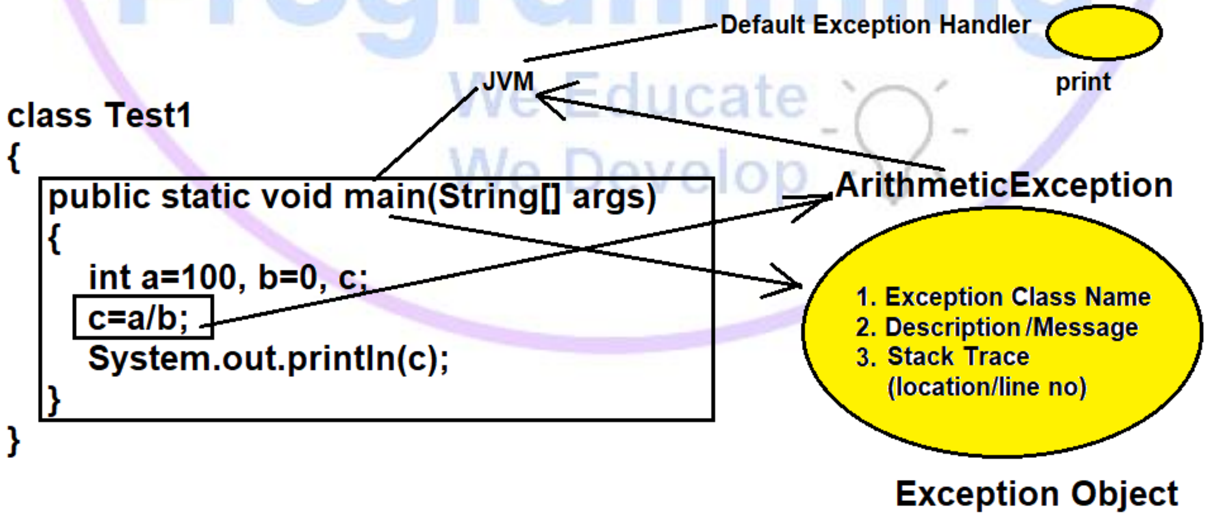
---

=> Working of exception :-

- ➔ If we do not handle the exception, out program will terminate abnormally (which is not good programming)
- ➔ If any exception is occured in any method, then that method will create an exception object and

that method has the responsibility to handle the exception

- If the method is not handling the exception, then caller method has the responsibility to handle the exception
- If any caller method is not handling the exception object then that object will be transfered to JVM and then Default Exception Handler
- Then Default Exception Handler will print the exception object and JVM will terminate the exception object abnormally



```
class Test1
{
 public static void main(String[] args)
 {
   int a=100, b=0, c;
   c=a/b;
   System.out.println(c);
 }
}
```

Default Exception Handler
print
JVM
ArithmeticException
1. Exception Class Name
2. Description /Message
3. Stack Trace
(location/line no)
Exception Object

=> throw keyword :-

- ➔ throw keyword is used to create an exception object programmatically
- ➔ Syntax :
  - throw new ExceptionClassName();

- ➔ throw keyword is mainly used for User-Defined Exceptions

=> How to handle the exceptions :-

-> We can handle the exception by 2 ways :-

1. by using "throws" keyword
2. by using "try-catch" block

Company Links & Contacts

Company Name: Smart Programming (+91 62838-30308)

Address : Chandigarh & Mohali (Punjab), India

Websites: https://www.smartprogramming.in/

https://courses.smartprogramming.in

Android App:
https://play.google.com/store/apps/details?id=com.smartprogramming

YouTube Channel:
https://www.youtube.com/c/SmartProgramming

Smart Programming : YouTube Channel

An investment in Knowledge pays the best interest….

Smart Programming
We Educate - We Develop

☎ +91 62838-30308
Call us to Learn
Latest Technologies

City : Mohali (Punjab),
& Chandigarh
(India)

WEBSITE : http://www.smartprogramming.in

BUY COURSES ON : https://courses.smartprogramming.in

YOUTUBE CHANNEL : Smart Programming (https://www.youtube.com/c/SmartProgramming)

ANDROID APP : Smart Programming
(https://play.google.com/store/apps/details?id=com.smartprogramming)

https://www.facebook.com/smartprogramming.india

https://www.instagram.com/smart_programming

Java
android
python
Data Science
Machine Learning
Artificial Intelligence
M E A N
Full Stack
BIG DATA
hadoop

"throws" keyword & "try-catch" block in Java

=> throws keyword :-

- ➔ throws keyword is used to inform the caller method that this method can throw one of the listed type of exceptions.
- ➔ throws keyword bypass the generated exceptions from present method to caller method
- ➔ If caller method wants to handle the exceptions then it has to use try-catch block
- ➔ NOTE : throws keyword cannot handle the exceptions

- ➔ Syntax : throws ExceptionClassName, ExceptionClassName,-,-

(it is used with method signature)

=> What is difference between throw & throws keyword ?

- ➔ "throw" keyword is used to create an exception object manually
  "throws" keyword is used to inform that this method can raise listed type of exceptions

- ➔ "throw" keyword is used in method body
  "throws" keyword is used with method signature

- ➔ "throw" keyword is able to allow only one exception class name at one time
  "throws" keyword is able to allow more than one exception class name at one time

- ➔ "throw" keyword is mainly used for unchecked exceptions
  "throws" keyword is mainly used for checked exceptions

➔ "throw" keyword is followed by an instance
"throws" keyword is followed by class

=> try-catch block :-

-> try block :- try block is used to provide the risky code i.e the code in which there are chances for exception

-> Syntax of try block :

```
try
{


}
```

-> catch block :- catch block is used ot provide maintenance code or alternative code to handle the exceptions or simply say exception handling code

-> Syntax of catch block :

```
catch(ExceptionClassName e)
{


}
```

➔ **try-catch flow :-**

= try block will be executed completely if there is no exception in try block. But if there is exception in try block then after that exception line it will skip the below code

= catch block will be executed only when there is an exception in try block. If there is no exception in try block then catch block will not execute.

= We can use multiple catch blocks with one try block

=> Different ways to print exception :-

1. System.out.println(e.getMessage()); //will display only message

2. System.out.println(e); //will display exception class name and message

3. e.printStackTrace(); //will display exception class name, message and line numbers

---

Company Links & Contacts

Company Name: Smart Programming (+91 62838-30308)

Address : Chandigarh & Mohali (Punjab), India

Websites: https://www.smartprogramming.in/

https://courses.smartprogramming.in

Android App: https://play.google.com/store/apps/details?id=com.smartprogramming

YouTube Channel: https://www.youtube.com/c/SmartProgramming

Smart Programming : YouTube Channel

An investment in Knowledge pays the best interest….

Smart Programming

We Educate - We Develop

+91 62838-30308

Call us to Learn Latest Technologies

City : Mohali (Punjab), & Chandigarh (India)

WEBSITE : http://www.smartprogramming.in

BUY COURSES ON : https://courses.smartprogramming.in

YOUTUBE CHANNEL : Smart Programming (https://www.youtube.com/c/SmartProgramming)

ANDROID APP : Smart Programming (https://play.google.com/store/apps/details?id=com.smartprogramming)

https://www.facebook.com/smartprogramming.india

https://www.instagram.com/smart_programming

Java
android
python
Data Science
Machine Learning
Artificial Intelligence
M E A N
Full Stack
BIG DATA
hadoop

“finally” block in Java

=> finally block :-

- ➔ finally is the block which is always executes after try or catch block
- ➔ finally block contains resource closing code for example database connection, reading or writing from file etc

=> Java 7 Updates for Exception Handling :-

1. Multi-catch block :

- ➔ Before Java 7 version, we can catch only one exception type in each catch block. So whenever we needed to handle more than one specific exception, but take same action for all exceptions, then we had to have more than one catch block containing the same code.
- ➔ But from Java 7.0 version, it is possible for a single catch block to catch multiple exceptions by separating each with | (pipe symbol) in catch block.
- ➔ Syntax :
  Catch(Exception1 | Exception2 | - | e) {body}

2. try with resource :-

- ➔ try with resource is used to automatically close the resources that the programmer has created
- ➔ try with resource can contain only that class which inherits AutoCloseable interface
- ➔ Due to "try with resource", the importance of "finally" block has decreased
- ➔ Syntax :

```
try(Resourse1; Resourse2; -; -)
{ body }
catch(ExceptionClassName e)
{ body }
```

Company Links & Contacts

Company Name: Smart Programming (+91 62838-30308)

Address : Chandigarh & Mohali (Punjab), India

Websites: https://www.smartprogramming.in/
https://courses.smartprogramming.in

Android App:
https://play.google.com/store/apps/details?id=com.smartprogramming

YouTube Channel:
https://www.youtube.com/c/SmartProgramming

Smart Programming : YouTube Channel

An investment in Knowledge pays the best interest....

Smart Programming

We Educate - We Develop

+91 62838-30308

Call us to Learn
Latest Technologies

City : Mohali (Punjab),
& Chandigarh
(India)

WEBSITE : http://www.smartprogramming.in

BUY COURSES ON : https://courses.smartprogramming.in

YOUTUBE CHANNEL : Smart Programming (https://www.youtube.com/c/SmartProgramming)

ANDROID APP : Smart Programming
(https://play.google.com/store/apps/details?id=com.smartprogramming)

https://www.facebook.com/smartprogramming.india

https://www.instagram.com/smart_programming

Java
android
python
Data Science
Machine Learning
Artificial Intelligence
M E A N
Full Stack
BIG DATA
hadoop

Final, finally and finalize; User Defined Exceptions in Java

=> try-catch-finally combinations :-

➔ Note : We can use either catch or finally or both with try block

=> What is difference between final, finally & finalize ?

-> "final" is the keyword

"finally" is the block

"finalize" is the method

-> "final" keyword can be used with variable (the value cannot be changed or re-assinged the final variable), method (we cannot override the final method) & class (we cannot inherit the final class)

"finally" block is used to close the resources

"finalize" is the Object class method which is executed just prior to the garbage collector (or gc()

method). It is used to perform the clean-up activity related to the object.

=> User Defined Exceptions :-

- ➔ User defined exceptions can be Checked or Unchecked exceptions
- ➔ While we create an API we normally create Checked Exceptions
- ➔ When we want to create normal exception then we can use Runtime Exceptions
- ➔ To create Checked Exception we have to use "Exception" class
- ➔ To create Unchecked Exceptions we have to use "RuntimeException" class

-> Steps to create User Defined Exceptions :-

1. Create a user defined exception class and inherit Exception or RuntimeException class
2. Create a parametrized constructor
3. "throw" the exception and handle it

Company Links & Contacts

Company Name: Smart Programming (+91 62838-30308)

Address : Chandigarh & Mohali (Punjab), India

Websites: https://www.smartprogramming.in/

https://courses.smartprogramming.in

Android App: https://play.google.com/store/apps/details?id=com.smartprogramming

YouTube Channel: https://www.youtube.com/c/SmartProgramming

Smart Programming : YouTube Channel

An investment in Knowledge pays the best interest….

Smart Programming
We Educate - We Develop

☎ +91 62838-30308
Call us to Learn
Latest Technologies

City : Mohali (Punjab), & Chandigarh (India)

WEBSITE : http://www.smartprogramming.in

BUY COURSES ON : https://courses.smartprogramming.in

YOUTUBE CHANNEL : Smart Programming (https://www.youtube.com/c/SmartProgramming)

ANDROID APP : Smart Programming (https://play.google.com/store/apps/details?id=com.smartprogramming)

https://www.facebook.com/smartprogramming.india

https://www.instagram.com/smart_programming

Java
android
python
Data Science
Machine Learning
Artificial Intelligence
M E A N
Full Stack
BIG DATA
hadoop

String Introduction in Java

=> String :-

- String is the sequence of characters or say String is an array of characters. For example : char[] c={'d', 'e', 'e', 'p', 'a', 'k'};
  - To handle the character sequence, java has provided an interface i.e. CharSequence

- String is a non-primitive data type.

- To create String or to perform String operations, java has provided some predefined classes :-

1. java.lang.String
2. java.lang.StringBuffer
3. java.lang.StringBuilder
4. java.util.StringTokenizer

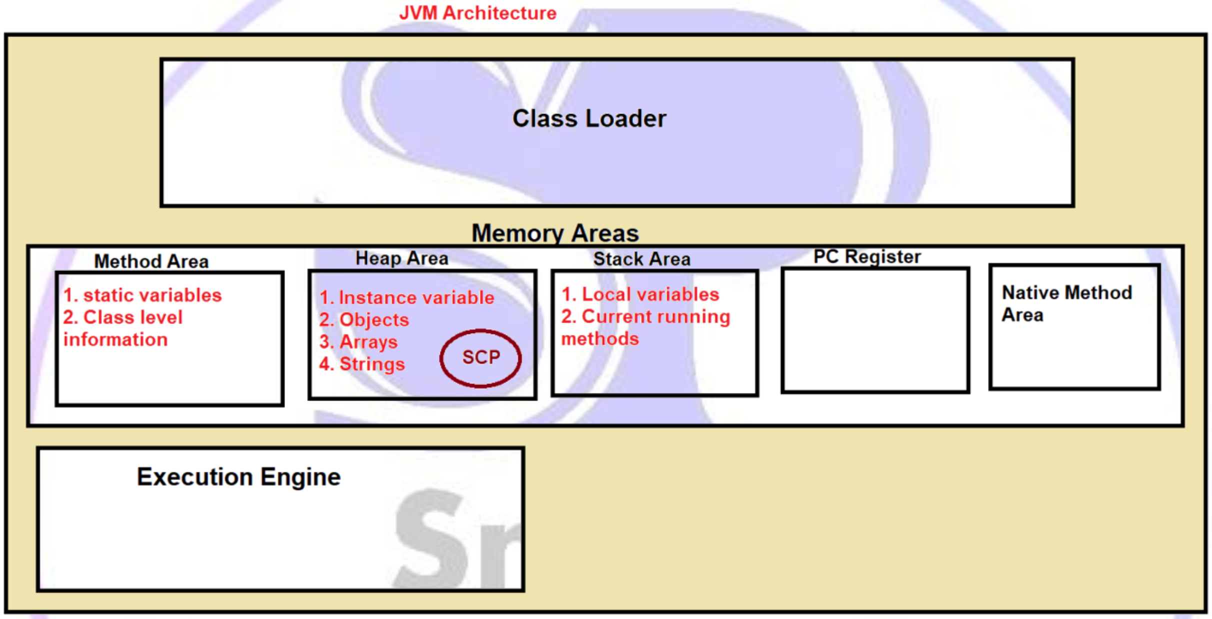
- ➔ Syntax :

```
public final class String extends Object implements Serializable, Comparable, CharSequence
{
    //methods
}
```

- ➔ String is the class and we can create String class object. But we can create String class object by 2 ways :-
  1. String str=new String("deepak");
  2. String str="deepak";

- ➔ Whenever we create String class object, objects created are "IMMUTABLE"

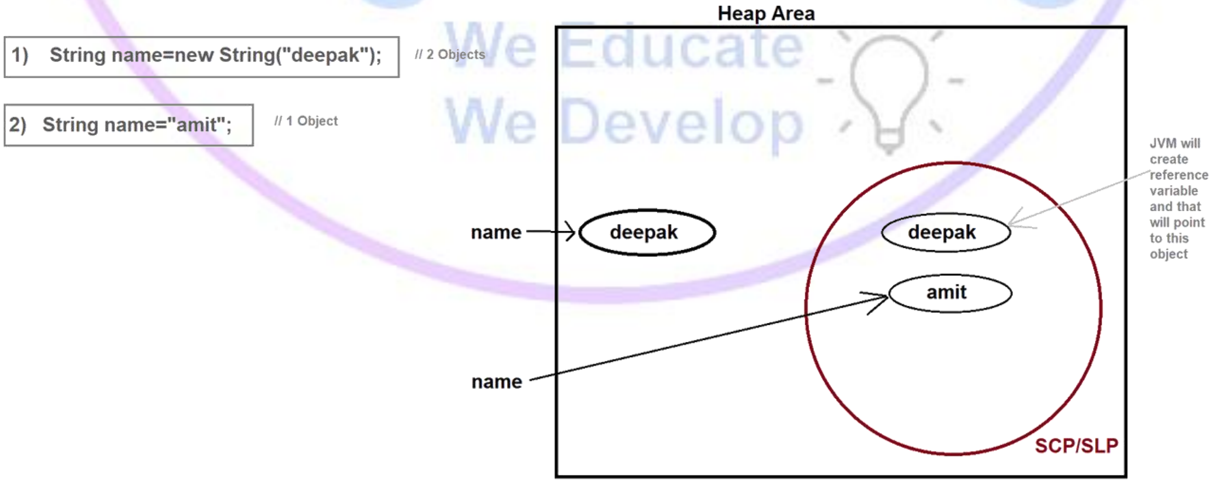
➔ Whenever we create String objects, it allocates memory in special memory area i.e. "String Constant Pool" or "String Literal Pool"


JVM Architecture
Class Loader
Memory Areas
Method Area
1. static variables
2. Class level information
Heap Area
1. Instance variable
2. Objects
3. Arrays
4. Strings
SCP
Stack Area
1. Local variables
2. Current running methods
PC Register
Native Method Area
Execution Engine

➔ Garbage collection is not appolicable for String Constant Pool

=> Difference between creating String objects by "new keyword" and by "String literal"

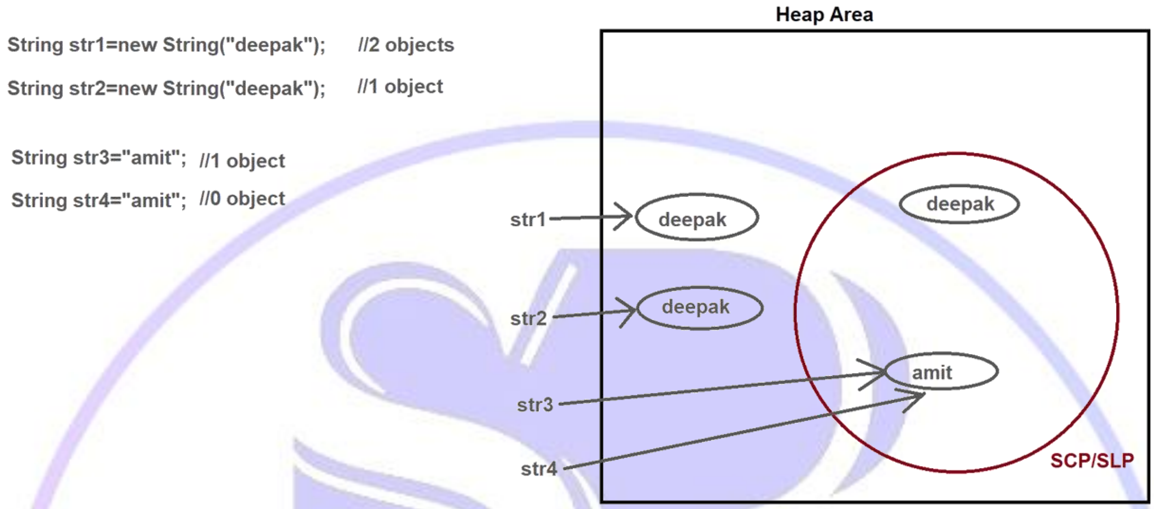
- ➔ If we create String object by using new keyword then an object is created in heap area. If we have provided any string literal in string constructor then 2 objects will be created and second object will be created in String Constant Pool
- ➔ If we create String object by using String literal then an object is created in String Constant Pool



1) String name=new String("deepak"); // 2 Objects

2) String name="amit"; // 1 Object

Heap Area
name
deepak
deepak
amit
name
SCP/SLP
JVM will create reference variable and that will point to this object



```
String str1=new String("deepak");    //2 objects
String str2=new String("deepak");    //1 object

String str3="amit";  //1 object
String str4="amit";  //0 object
```

=> Properties of "String Constant Pool" or "String Literal Pool"

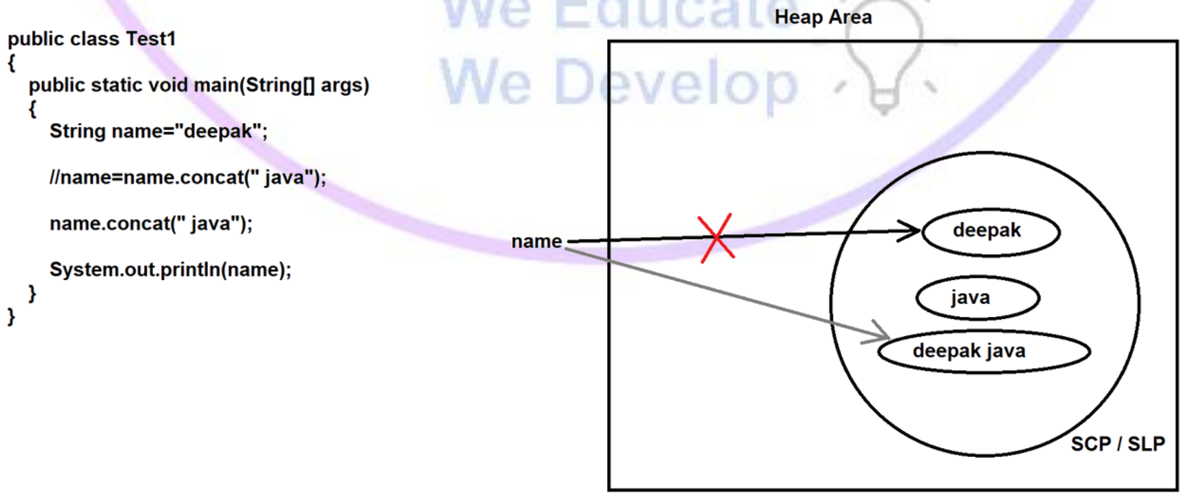
1. SCP stores the String Literal Objects

2. Whenever String Literal Object is created in SCP, first it will check wether that literal object is already present in SCP or not, if it is not present then it will create new object otherwise it will not create new object and that reference variable will point to that same object

3. Garbage Collection is not applicable in SCP

=> Why String objects are immutable ?

- ➔ String objects are immutable which means that if we create any String object, then we cannot change its value
- ➔ String objects are immutable because if we change any string object value, then it will create new object and will not affect other reference variables
- ➔ NOTE : Strings are not immutable, String objects are immutable

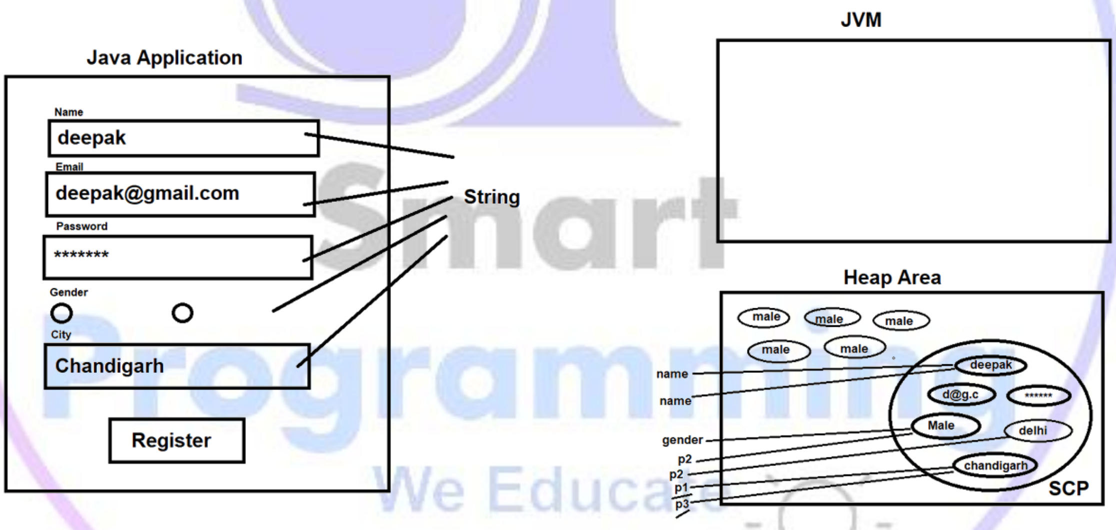


```
public class Test1
{
   public static void main(String[] args)
   {
      String name="deepak";

      //name=name.concat(" java");

      name.concat(" java");

      System.out.println(name);
   }
}
```

=> What String class is "final" ?

-> String class is final because we cannot inherit String class and thus we were not able to change the String class properties

Real world use of String final class and string immutable object


Java Application
Name
deepak
Email
deepak@gmail.com
Password
****
Gender
City
Chandigarh
Register
String
JVM
Heap Area
male
male
male
male
male
deepak
d@g.c
******
Male
delhi
chandigarh
SCP
name
name
gender
p2
p2
p1
p3

Company Links & Contacts

Company Name: Smart Programming (+91 62838-30308)

Address : Chandigarh & Mohali (Punjab), India

Websites: https://www.smartprogramming.in/

https://courses.smartprogramming.in

Android App: https://play.google.com/store/apps/details?id=com.smartprogramming

YouTube Channel: https://www.youtube.com/c/SmartProgramming

Smart Programming : YouTube Channel

An investment in Knowledge pays the best interest….

Smart Programming
We Educate - We Develop

+91 62838-30308
Call us to Learn
Latest Technologies

City : Mohali (Punjab),
& Chandigarh
(India)

WEBSITE : http://www.smartprogramming.in

BUY COURSES ON : https://courses.smartprogramming.in

YOUTUBE CHANNEL : Smart Programming (https://www.youtube.com/c/SmartProgramming)

ANDROID APP : Smart Programming
(https://play.google.com/store/apps/details?id=com.smartprogramming)

https://www.facebook.com/smartprogramming.india

https://www.instagram.com/smart_programming

Java
ANDROID
python
Data Science
Machine Learning
Artificial Intelligence
M E A N
Full Stack
BIG DATA
hadoop

String class methods in Java

=> String class Methods :-

➔ Validating User Input

- length() - It provides the int value i.e. no of characters in the String
- isEmpty() - Checks wether string is empty or not, if empty it will return true else false
- trim() - It is used to remove the front or back side spaces

➔ Comparing two strings :-

- equals() - it compares the 2 strings and retruns the boolean value
- equalsIgnoreCase() - it compares the 2 strings but ignores the uppercase and lowercase
- compareTo() - It compares the 2 strings lexicographically
- compareToIgnoreCase()

➔ Contination of two strings :-

- '+' operator
- concat() - It concats the 2 strings

➔ Get sub-string from String :-

- subString() - It will return the sub string according to the provided starting and ending index positions
- subSequence() - It will return the CharSequence according to the provided starting and ending index positions

➔ Replacing or Removing characters :-

- replace() - It will replace the string according to provided another string
- replaceFirst() - It will replace the first string
- replaceAll() - it will replace all the strings

➔ Searching characters in string :-

- indexOf() - It will retrun the index position of provided character
- lastIndexOf() - It will retrun the index position of provided character from last position
- contains() - It will return boolean value, true if string present else false
- charAt() - It will return the index position of provided character
- endsWith() - It will return boolean value if string is matched from ending
- startsWith()- It will return boolean value if string is matched from starting

➔ Case conversion methods :-

- toLowerCase() - It will convert the string into lowercase
- toUpperCase() - It will convert the string into uppercase

➔ Type conversion methods :-

- valueOf() - It will convert other data type into string (it is static method)
- toCharArray() - It will convert the string into character array

➔ Other method :

- split() - It will split the string according to provided regex

Company Links & Contacts

Company Name: Smart Programming (+91 62838-30308)

Address : Chandigarh & Mohali (Punjab), India

Websites: https://www.smartprogramming.in/

https://courses.smartprogramming.in

Android App: https://play.google.com/store/apps/details?id=com.smartprogramming

YouTube Channel: https://www.youtube.com/c/SmartProgramming

Smart Programming : YouTube Channel

An investment in Knowledge pays the best interest….

Smart Programming

We Educate - We Develop

☎ +91 62838-30308

Call us to Learn Latest Technologies

City : Mohali (Punjab), & Chandigarh (India)

WEBSITE : http://www.smartprogramming.in

BUY COURSES ON : https://courses.smartprogramming.in

YOUTUBE CHANNEL : Smart Programming (https://www.youtube.com/c/SmartProgramming)

ANDROID APP : Smart Programming (https://play.google.com/store/apps/details?id=com.smartprogramming)

https://www.facebook.com/smartprogramming.india

https://www.instagram.com/smart_programming

Java
android
python
Data Science
Machine Learning
Artificial Intelligence
M E A N
Full Stack
BIG DATA
hadoop

StringBuffer, StringBuilder & StringTokenizer class methods in Java

=> What is difference between equals() and == ?

-> NOTE :

1. equals() method is the "Object" class method. Object class equals() method compare the reference of 2 objects (address comparison)
2. String class overrides the equals() method of Object class

- ➔ == operator is used for reference comparison or address comparison
- ➔ equals() method is used for content comparison

=> StringBuffer class :-

- ➔ In some cases using String objects is not helpful because String objects are immutable but we want to update the same string object. If we want to update the string object again and again then java has provided one class i.e. StringBuffer class
- ➔ NOTE : In case of StringBuffer string mutable object is created

- ➔ StringBuffer is the class thus it has some constructors and methods

=> StringBuilder class :-

➔ StringBuilder is same as StringBuffer but one difference is that all the methods of StringBuilder are non-synchronized but all the methods of StringBuffer are synchronized

=> What is difference between StringBuffer & StringBuilder :-

1. StringBuffer methods are synchronized
   StringBuilder methods are non-synchronized

1. StringBuffer follows the sequential execution
   StringBuilder follows the parallel execution

2. StringBuffer will take more execution time
   StringBuilder will take less execution time

3. In case of StringBuffer application performance is slow
   In case of StringBuilder application performance is fast

4. StringBuffer is threadsafe
   StringBuilder is not threadsafe

5. StringBuffer provides the guarantee for data consistency
   StringBuilder does not provides the guarantee for data consistency

6. StringBuffer came in JDK 1.0 version
   StringBuilder came in JDK 1.5 version

=> StringTokenizer :-

- ➔ StringTokenizer is the class which is used to divide the strings into tokens
- ➔ Methods :-

1. hasMoreTokens()
2. nextToken()
3. countTokens()

4. hasMoreElements()
5. nextElement()

Company Links & Contacts

Company Name: Smart Programming (+91 62838-30308)

Address : Chandigarh & Mohali (Punjab), India

Websites: https://www.smartprogramming.in/

https://courses.smartprogramming.in

Android App: https://play.google.com/store/apps/details?id=com.smartprogramming

YouTube Channel: https://www.youtube.com/c/SmartProgramming

Smart Programming : YouTube Channel

An investment in Knowledge pays the best interest….

Smart Programming

We Educate - We Develop

+91 62838-30308

Call us to Learn
Latest Technologies

City : Mohali (Punjab),
& Chandigarh
(India)

WEBSITE : http://www.smartprogramming.in

BUY COURSES ON : https://courses.smartprogramming.in

YOUTUBE CHANNEL : Smart Programming (https://www.youtube.com/c/SmartProgramming)

ANDROID APP : Smart Programming
(https://play.google.com/store/apps/details?id=com.smartprogramming)

https://www.facebook.com/smartprogramming.india

https://www.instagram.com/smart_programming

Java
android
python
Data Science
Machine Learning
Artificial Intelligence
M E A N
Full Stack
BIG DATA
hadoop

Multithreading Introduction

in Java

=> Multitasking :-

- ➔ Performing multiple task at single time
- ➔ Examples :
  - o When we perform multiple task for example opening vlc, word, notepad, browser on single system etc
- ➔ Multitasking use the concept of context switching internally
- ➔ Multitasking can be achieved by 2 ways :-

1. Process based multitasking (Multiprocessing)
2. Thread based multitasking (Multithreading)

- ➔ Multitasking is used to reduce the ideal time of CPU.

=> Multiprocessing :-

- ➔ Multiprocessing is the part when one system is connected with multiple processor (CPU)
- ➔ Multiprocessing is best suitable at system level or OS level

=> Multithreading :-

- ➔ Multithreading is the part which which multiple threads (sub-process) are executed at single time
- ➔ Examples :

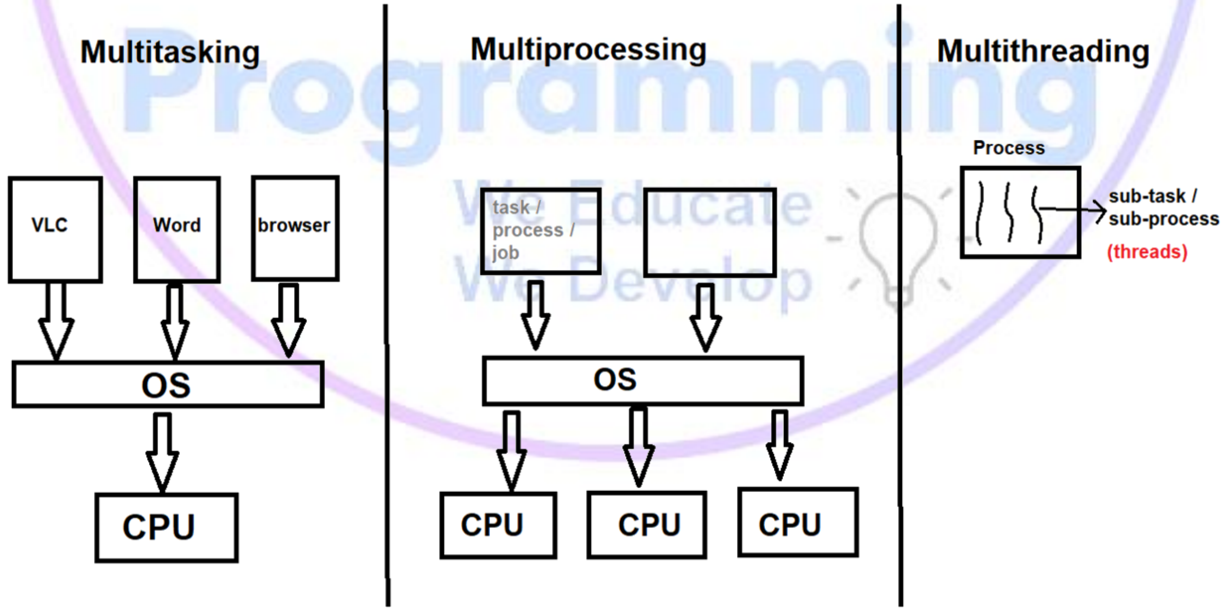
1. Word
2. VLC
3. Games
4. Animations
5. Browser
6. Web-servers

etc

➔ Multithreading is mostly implemented by java itself, we have to implement less part of multithreading.

➔ Java provides API for multithreading :-

1. Thread
2. Runnable
3. ThreadGroup
4. Concureency
5. ThreadPool
1. etc


Multitasking
VLC
Word
browser
OS
CPU
Multiprocessing
task / process / job
OS
CPU
CPU
CPU
Multithreading
Process
sub-task / sub-process
(threads)

=> What is difference between Process & Thread ?

1. Process is the heavy-weight task performing multiple tasks
   Thread is the light-weight sub-process performing single task

2. Process does not depends on each other
   Threads depends on each other

3. Context switching is difficult in process (takes more time)
   Context switching is easy in threads (takes less time)

3. Process shares the different address space
   Threads shares the same address space

4. Process does not require synchronization
   Threads requires synchronization

=> How to create threads :

-> There are 2 ways to create threads :-

1. By using "Thread" class
2. By using "Runnable" interface

=> What is "Thread" :-

➔ Thread is the pre-defined class which is present in java.lang package

- Syntax :

```
class Thread implements Runnable
{
        //constructors


        //methods
        run()
        start()
        etc
}
```

=> What is "Runnable" :-

- Runnable is pre-defined interface present in java.lang package

➔ Syntax :

```
interface Runnable
{
        //method
        run()
}
```

=> Which is better way to create thread, Thread or Runnable ?

-> Runnable is better way to create threads in java because if we inherit Thread class then we were not able to inherit any other class as multiple inheritance is not supported in java but if we inherit Runnable interface then we can inherit more interfaces and can inherit the class also

Company Links & Contacts

Company Name: Smart Programming (+91 62838-30308)

Address : Chandigarh & Mohali (Punjab), India

Websites: https://www.smartprogramming.in/

https://courses.smartprogramming.in

Android App: https://play.google.com/store/apps/details?id=com.smartprogramming

YouTube Channel: https://www.youtube.com/c/SmartProgramming

Smart Programming : YouTube Channel

An investment in Knowledge pays the best interest....

Smart Programming
We Educate - We Develop

☎ +91 62838-30308
Call us to Learn
Latest Technologies

City : Mohali (Punjab), & Chandigarh (India)

WEBSITE : http://www.smartprogramming.in

BUY COURSES ON : https://courses.smartprogramming.in

YOUTUBE CHANNEL : Smart Programming (https://www.youtube.com/c/SmartProgramming)

ANDROID APP : Smart Programming (https://play.google.com/store/apps/details?id=com.smartprogramming)

https://www.facebook.com/smartprogramming.india

https://www.instagram.com/smart_programming

Java
android
python
Data Science
Machine Learning
Artificial Intelligence
M E A N
Full Stack
BIG DATA
hadoop

Creating Threads (Part 1)
in Java

=> By using "Thread" class :-

-> Steps to create thread using "Thread" class :-

1. inherits the "Thread" class
2. override the run() method
3. create an instance of the class
4. start the thread

==
==

=> What is the use of start() method :-

1. It register the thread with "thread scheduler"

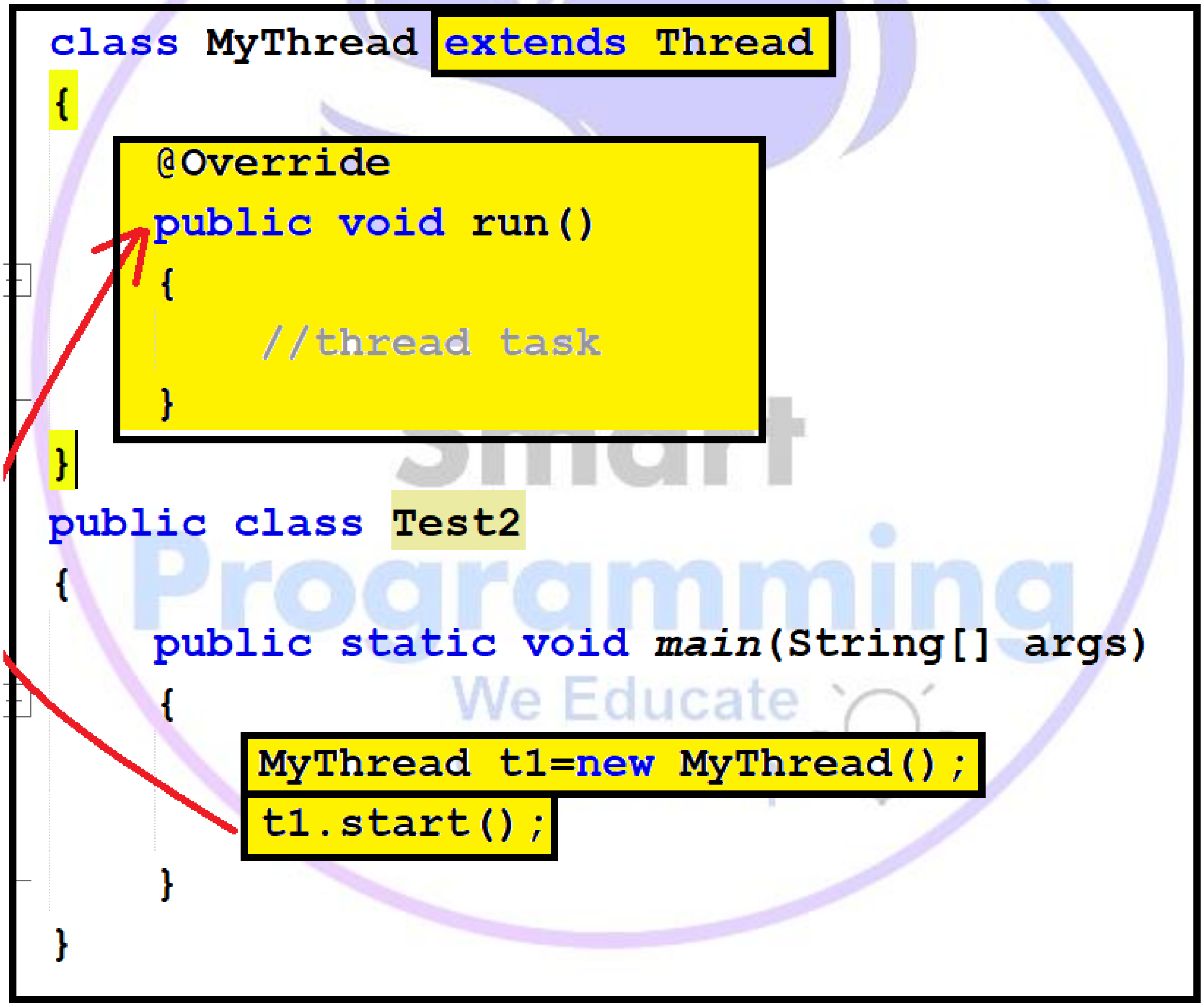
2. It will execute all the mentadory functionalities related to thread

3. It will invoke the run() method

etc

Diagrams used in the videos :-

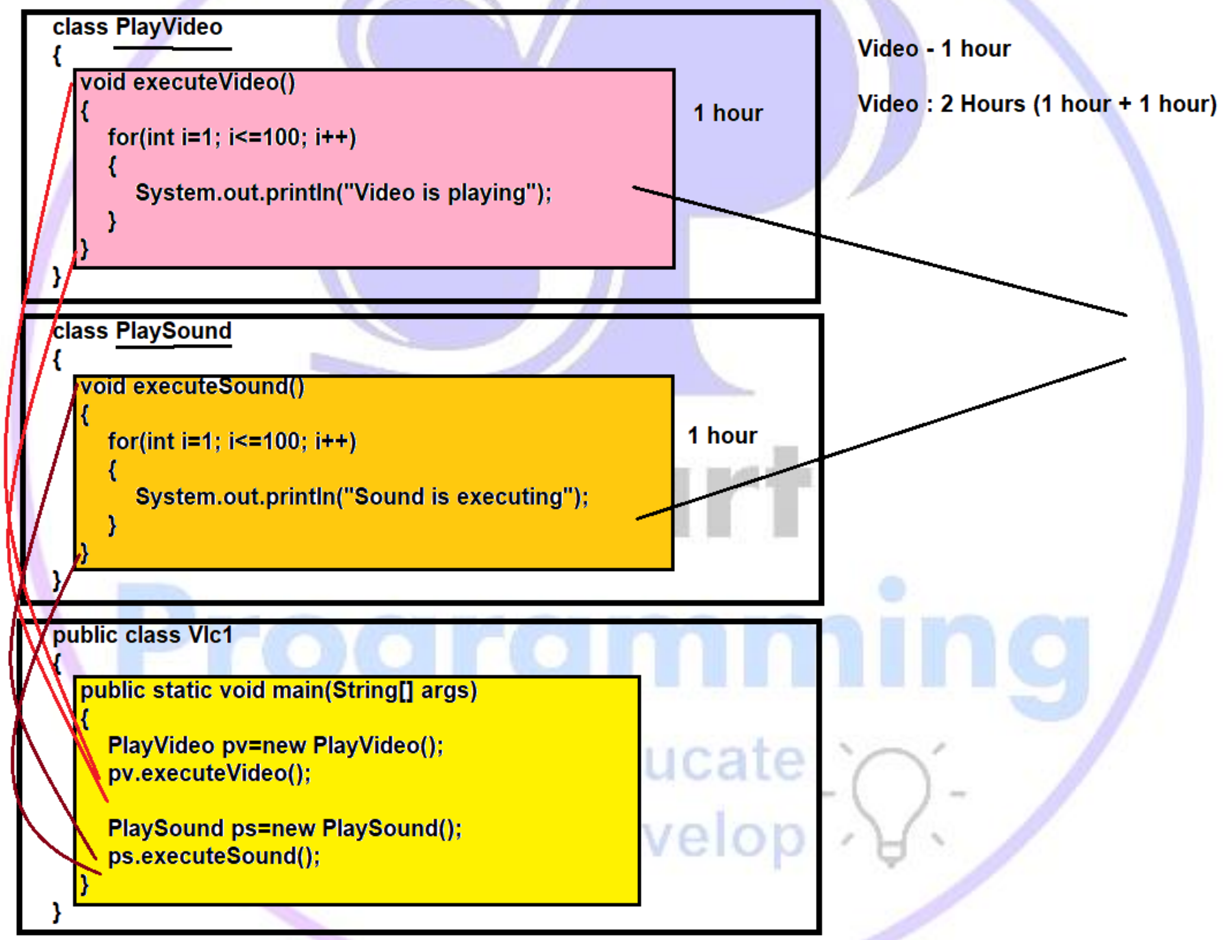
- ➔ Diagram 1 : (flow of simple thread)



```
class MyThread extends Thread
{
    @Override
    public void run()
    {
        //thread task
    }
}
public class Test2
{
    public static void main(String[] args)
    {
        MyThread t1=new MyThread();
        t1.start();
    }
}
```

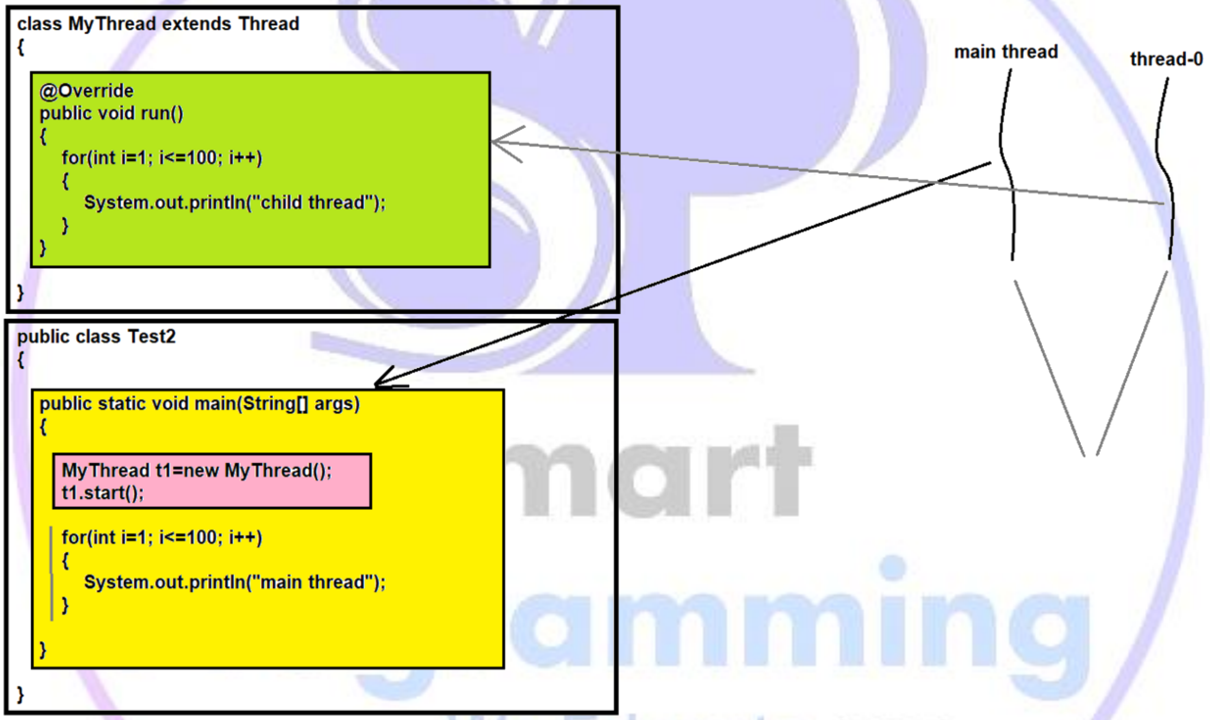
➔ Diagram 2 :

(If we don’t use thread concept then time taken to perform task will increase)



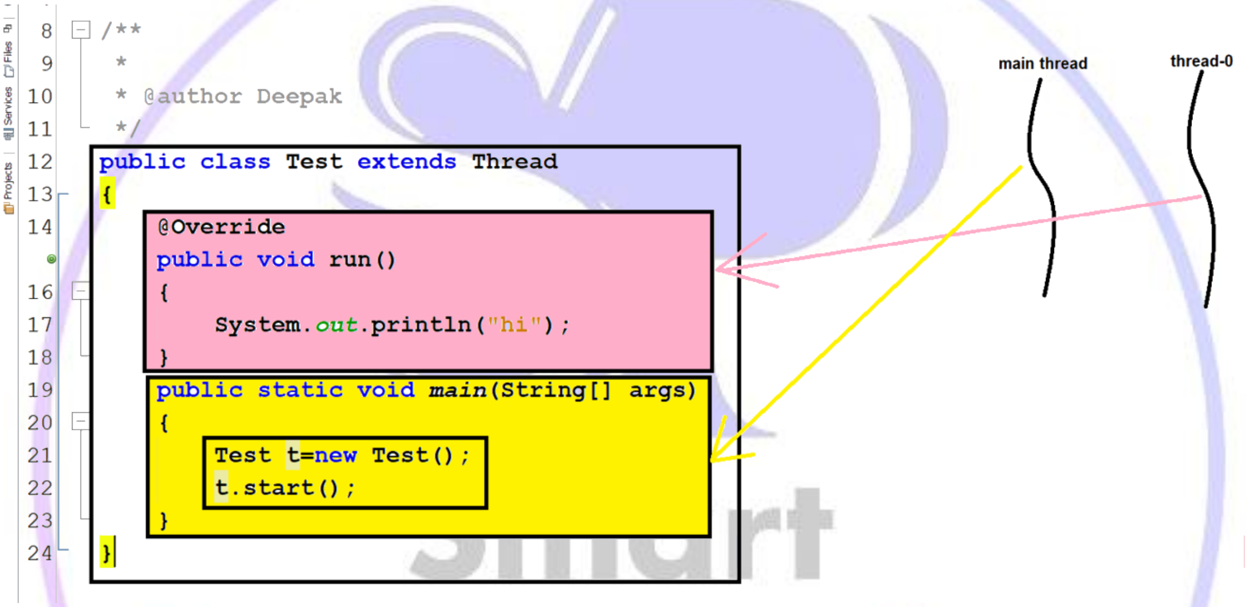
```
class PlayVideo
{
    void executeVideo()
    {
        for(int i=1; i<=100; i++)
        {
            System.out.println("Video is playing");
        }
    }
}
class PlaySound
{
    void executeSound()
    {
        for(int i=1; i<=100; i++)
        {
            System.out.println("Sound is executing");
        }
    }
}
public class Vlc1
{
    public static void main(String[] args)
    {
        PlayVideo pv=new PlayVideo();
        pv.executeVideo();

        PlaySound ps=new PlaySound();
        ps.executeSound();
    }
}
```

➔ Diagram 3 :
(Main thread and user thread creation and their execution)



```
class MyThread extends Thread
{
    @Override
    public void run()
    {
        for(int i=1; i<=100; i++)
        {
            System.out.println("child thread");
        }
    }
}
public class Test2
{
    public static void main(String[] args)
    {
        MyThread t1=new MyThread();
        t1.start();

        for(int i=1; i<=100; i++)
        {
            System.out.println("main thread");
        }

    }
}
```

➔ Diagram 4 :
(Which thread will execute which part)



```
/**
 *
 * @author Deepak
 */
public class Test extends Thread
{
    @Override
    public void run()
    {
        System.out.println("hi");
    }
    public static void main(String[] args)
    {
        Test t=new Test();
        t.start();
    }
}
```

Company Links & Contacts

Company Name: Smart Programming (+91 62838-30308)

Address : Chandigarh & Mohali (Punjab), India

Websites: https://www.smartprogramming.in/

https://courses.smartprogramming.in

Android App: https://play.google.com/store/apps/details?id=com.smartprogramming

YouTube Channel: https://www.youtube.com/c/SmartProgramming

Smart Programming : YouTube Channel

An investment in Knowledge pays the best interest….

Smart Programming
We Educate - We Develop

+91 62838-30308
Call us to Learn Latest Technologies

City : Mohali (Punjab), & Chandigarh (India)

WEBSITE : http://www.smartprogramming.in

BUY COURSES ON : https://courses.smartprogramming.in

YOUTUBE CHANNEL : Smart Programming (https://www.youtube.com/c/SmartProgramming)

ANDROID APP : Smart Programming (https://play.google.com/store/apps/details?id=com.smartprogramming)

https://www.facebook.com/smartprogramming.india

https://www.instagram.com/smart_programming

Java
android
python
Data Science
Machine Learning
Artificial Intelligence
M E A N
Full Stack
BIG DATA
hadoop

Creating Threads (Part 2) in Java

=> By using "Runnable" interface :-

➔ Steps to create thread using "Runnable" interface :-

1. inherits the "Runnable" interface
2. override the run() method
3. create an instance of the class
4. create an instance of Thread class and pass the above class instance in thread class constructor
5. start the thread

=> Life cycle of thread :-

1. Born : New thread is created.

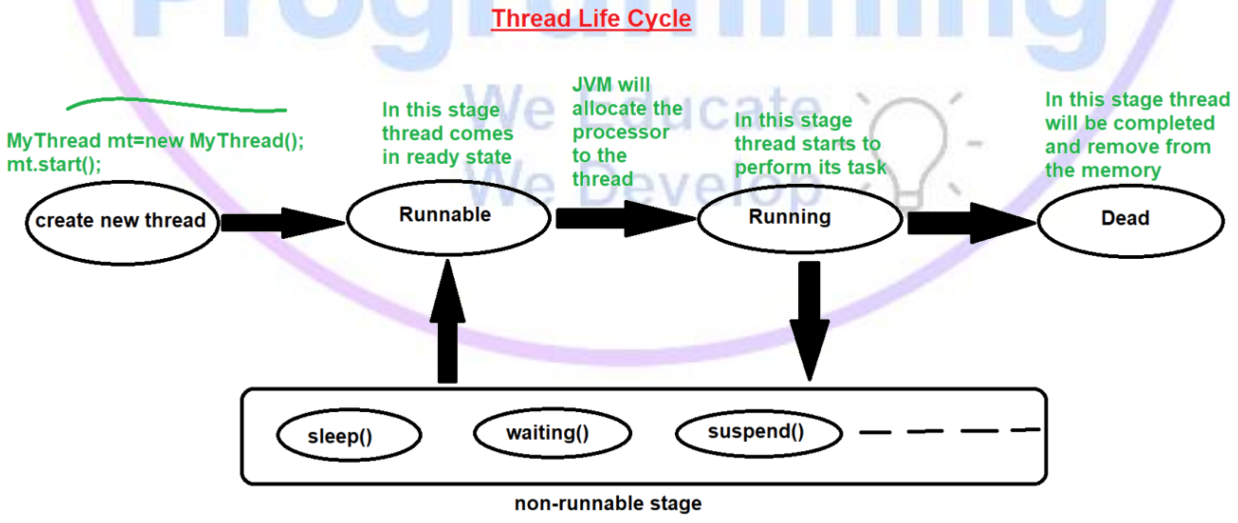
```
MyThread mt=new MyThread();
mt.start();
```

2. Runnable : In this stage thread comes in ready stage

3. Running : In this stage thread starts its task i.e. run() method is executing

4. Non-Runnable : In this stage thread does not perform any task. But when the thread is invoked then thread goes to the runnable stage and then it goes to the running stage

5. Dead : In this stage thread completes its task and removed from the memory


Thread Life Cycle
MyThread mt=new MyThread();
mt.start();
create new thread
In this stage thread comes in ready state
Runnable
JVM will allocate the processor to the thread
In this stage thread starts to perform its task
Running
In this stage thread will be completed and remove from the memory
Dead
sleep()
waiting()
suspend()
non-runnable stage

=> Note :

- ➔ We can envoke (start) the thread only once. If we try to start the thread again it will provide an exception saying java.lang.IllegalThreadStateException

---

Company Links & Contacts

Company Name: Smart Programming (+91 62838-30308)

Address : Chandigarh & Mohali (Punjab), India

Websites: https://www.smartprogramming.in/

https://courses.smartprogramming.in

Android App: https://play.google.com/store/apps/details?id=com.smartprogramming

YouTube Channel: https://www.youtube.com/c/SmartProgramming

Smart Programming : YouTube Channel

An investment in Knowledge pays the best interest….

Smart Programming

We Educate - We Develop

+91 62838-30308

Call us to Learn Latest Technologies

City : Mohali (Punjab), & Chandigarh (India)

WEBSITE : http://www.smartprogramming.in

BUY COURSES ON : https://courses.smartprogramming.in

YOUTUBE CHANNEL : Smart Programming (https://www.youtube.com/c/SmartProgramming)

ANDROID APP : Smart Programming (https://play.google.com/store/apps/details?id=com.smartprogramming)

https://www.facebook.com/smartprogramming.india

https://www.instagram.com/smart_programming

Java
android
python
Data Science
Machine Learning
Artificial Intelligence
M E A N
Full Stack
BIG DATA
hadoop

Thread class methods (Part 1) in Java

=> Thread class constructors :-

1. public Thread() { - }
2. public Thread(Runnable target) { - }
3. public Thread(String name) { - }
4. public Thread(Runnable target, String name) { - }
5. public Thread(ThreadGroup group, Runnable target) { - }
6. public Thread(ThreadGroup group, String name) { - }
7. public Thread(ThreadGroup group, Runnable target, String name) { - }
8. public Thread(ThreadGroup group, Runnable target, String name, long stackSize) { - }

=> Thread class methods :-

1. Simple methods :-

- ➔ run() - this method contains the thread task
- ➔ start() - this method is used to create thread

- ➔ currentThread() - this method returns the reference of current running thread
- ➔ isAlive() - this method is used to check wether the thread is running or not

2. Naming Methods :-

- ➔ getName() - this method is used to get the current running thread name
- ➔ setName(String name) - this method is used to set the name of current running thread

3. Daemon thread methods :

- ➔ isDaemon() - this method is used to check wether the thread is daemon thread or not
- ➔ setDaemon(boolean b) - this method is used to create daemon thread

=> Daemon Threads :-

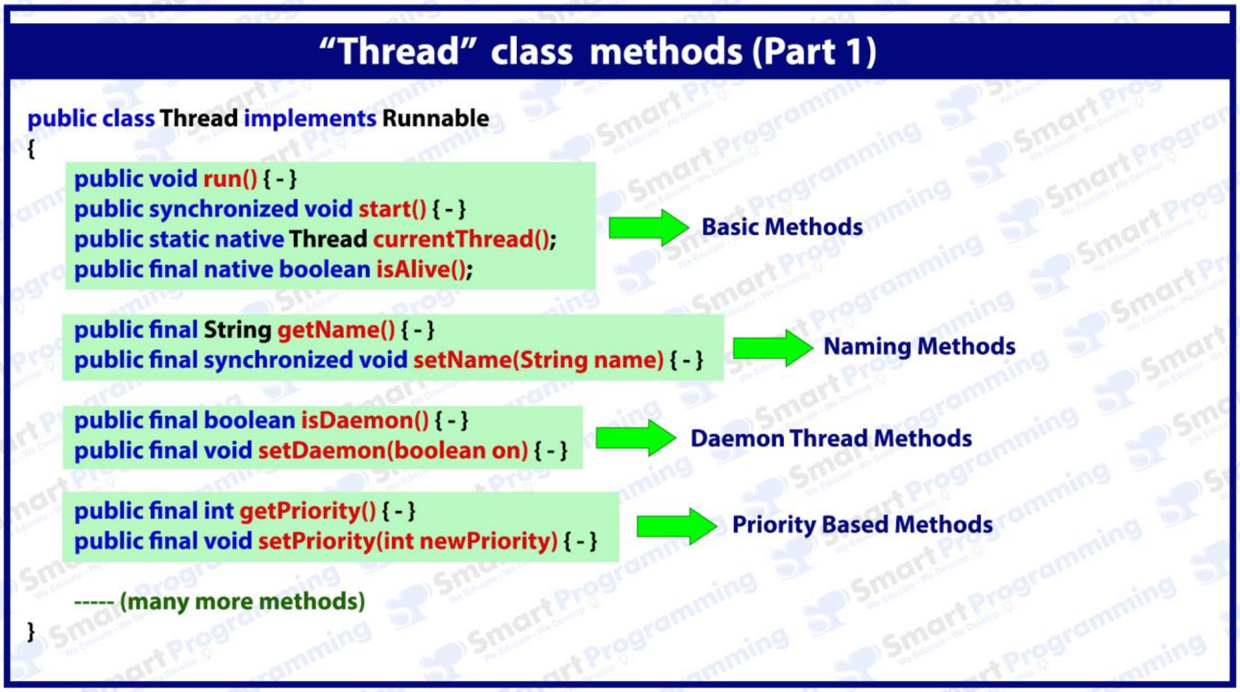
- ➔ Daemon threads are the threads which are executed in the background of another thread
- ➔ For example : Garbage collector, finalizer, spelling checker in word etc
- ➔ Daemon thread is used to provide the service to the other threads
- ➔ Daemon thread methods :
  - ○ isDaemon()
  - ○ setDaemon(boolean b)

➔ Note :

1. We cannot create the running thread as daemon thread. If we try to create daemon thread after thread has started then it will throw an exception saying "java.lang.IllegalThreadStateException"
2. We cannot create main thread as daemon thread because JVM starts the main thread before we create it as daemon thread

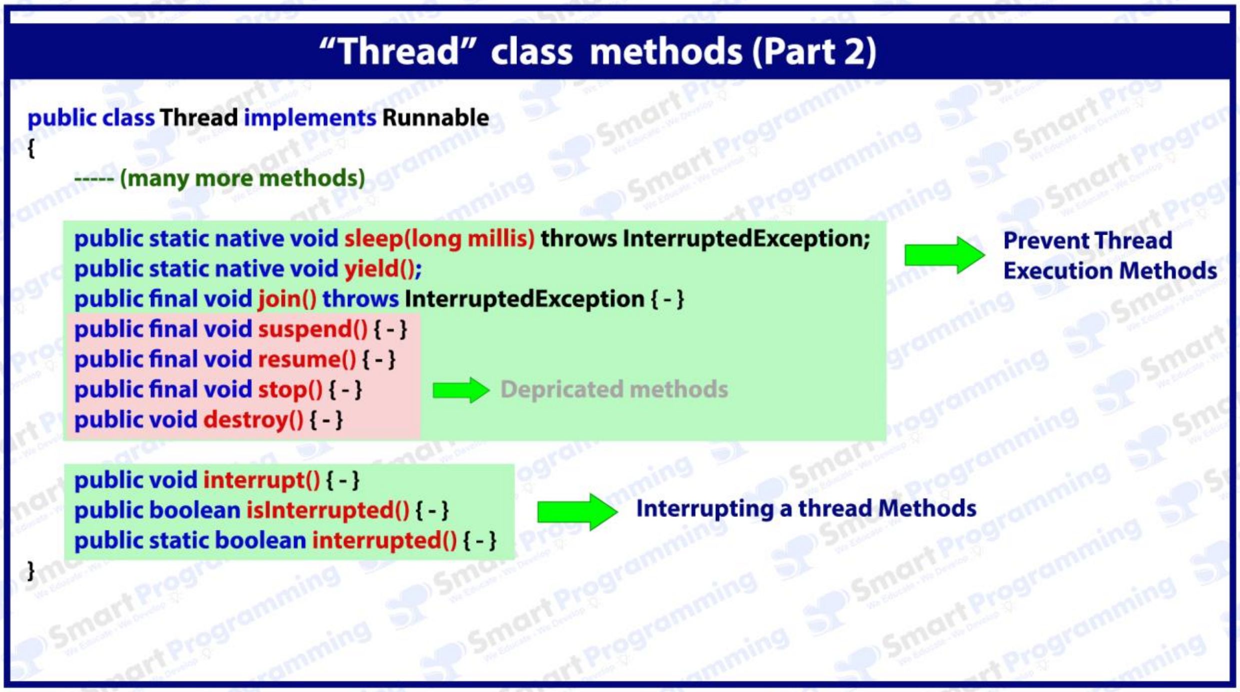
3. Daemon thread life depends on another thread in which it is running
4. Daemon thread inherits the properties/nature from its parent thread
5. JVM can stop the daemon thread but it cannot stop the normal thread
6. It is recommended to make the lowest priority of daemon thread

⇨ Thread class methods diagram :



"Thread" class methods (Part 1)

```
public class Thread implements Runnable
{
    public void run() { - }
    public synchronized void start() { - }
    public static native Thread currentThread();                  → Basic Methods
    public final native boolean isAlive();

    public final String getName() { - }
    public final synchronized void setName(String name) { - }     → Naming Methods

    public final boolean isDaemon() { - }
    public final void setDaemon(boolean on) { - }                 → Daemon Thread Methods

    public final int getPriority() { - }
    public final void setPriority(int newPriority) { - }          → Priority Based Methods

    ----- (many more methods)
}
```



"Thread" class methods (Part 2)

```
public class Thread implements Runnable
{
    ----- (many more methods)

    public static native void sleep(long millis) throws InterruptedException;   → Prevent Thread
    public static native void yield();                                             Execution Methods
    public final void join() throws InterruptedException { - }
    public final void suspend() { - }
    public final void resume() { - }
    public final void stop() { - }          → Depricated methods
    public void destroy() { - }

    public void interrupt() { - }
    public boolean isInterrupted() { - }                   → Interrupting a thread Methods
    public static boolean interrupted() { - }
}
```

Company Links & Contacts

Company Name: Smart Programming (+91 62838-30308)

Address : Chandigarh & Mohali (Punjab), India

Websites: https://www.smartprogramming.in/

https://courses.smartprogramming.in

Android App: https://play.google.com/store/apps/details?id=com.smartprogramming

YouTube Channel: https://www.youtube.com/c/SmartProgramming

Smart Programming : YouTube Channel

An investment in Knowledge pays the best interest….

Smart Programming

We Educate - We Develop

+91 62838-30308

Call us to Learn Latest Technologies

City : Mohali (Punjab), & Chandigarh (India)

WEBSITE : http://www.smartprogramming.in

BUY COURSES ON : https://courses.smartprogramming.in

YOUTUBE CHANNEL : Smart Programming (https://www.youtube.com/c/SmartProgramming)

ANDROID APP : Smart Programming (https://play.google.com/store/apps/details?id=com.smartprogramming)

https://www.facebook.com/smartprogramming.india

https://www.instagram.com/smart_programming

Java
android
python
Data Science
Machine Learning
Artificial Intelligence
M E A N
Full Stack
BIG DATA
hadoop

Thread class methods (Part 2) in Java

=> Thread class methods :-

4. Priority Based Methods :

- ➔ getPriority() - This method is used to get the priority of the thread
- ➔ setPriority(int priority) - This method is used to set the priority of the thread

5. Prevent thread execution method :-

- ➔ sleep(long mili) - This method is used to pause the current running thread for the provided time period
- ➔ yield() - This method is used to stop the current running thread execution and provide the chance to another thread for execution
- ➔ join() -

=> What is thread priority :-

- ➔ Thread priority is an integer value of the thread, the thread having high priority or high integer value will get the priority first to execute by JVM.
- ➔ Priority integer value ranges from 1 to 10.
- ➔ Java provides 3 pre-defined priorities :-
    - 1 - MIN_PRIORITY
    - 5 - NORM_PRIORITY
    - 10 - MAX_PRIORITY
- ➔ Below are not priorities :-
    - 0, <1, >10 (these priorities will provide an exception saying java.lang.IllegalArgumentException)
    - LOW_PRIORITY, MINIMUM_PRIORITY
    - NORMAL_PRIORITY, MIDIUM_PRIORITY
    - MAXIMUM_PRIORITY, HIGH_PRIORITY

➔ **NOTE :**

1. Priorities depends on the platform (Windows does not support thread priorities)
2. By default main thread has priority 5
3. Thread default priorities are inherited by parent thread (It also depends on the platform)
4. If multiple threads have same priority then which thread will get the chance to execute first depends on the JVM (thread schedular)

=> sleep() method :

- ➔ This method is used to pause the current running thread for the provided time period
- ➔ sleep() method is static method thus we have to call it by class name i.e. Thread class

- ➔ sleep() methods throws InterruptedException thus we have to use throws keyword or "try-catch block"
- ➔ We can provide time as 0 but it cannot be negative integer value (java.lang.IllegalArgumentException)
- ➔ JVM does not provide the guarantee that the sleeping thread will invoke exactly after provided time period
- ➔ When the thread goes into sleeping stage then it doesnt release the lock

---

=> yield() method :-

- ➔ This method stops the current thread execution and provide the chance to the other thread to execute
- ➔ Output is not constant in case of yield() method

➔ NOTE :-

- Till JDK 5 version, yield() method internally calls the sleep() method
- After JDK 5 version, yield() method working has been changed. It provides the hint to the thread schedular to yield(stop), but it totally depends on the thread-schedular that it will accept its request or not.
- If current running thread is stoped or yield, then which thread will get the chance for execution depends on the thread-schedular

Company Links & Contacts

Company Name: Smart Programming (+91 62838-30308)

Address : Chandigarh & Mohali (Punjab), India

Websites: https://www.smartprogramming.in/

https://courses.smartprogramming.in

Android App: https://play.google.com/store/apps/details?id=com.smartprogramming

YouTube Channel: https://www.youtube.com/c/SmartProgramming

Smart Programming : YouTube Channel

An investment in Knowledge pays the best interest….

Smart Programming

We Educate - We Develop

☎ +91 62838-30308

Call us to Learn Latest Technologies

City : Mohali (Punjab), & Chandigarh (India)

WEBSITE : http://www.smartprogramming.in

BUY COURSES ON : https://courses.smartprogramming.in

YOUTUBE CHANNEL : Smart Programming (https://www.youtube.com/c/SmartProgramming)

ANDROID APP : Smart Programming (https://play.google.com/store/apps/details?id=com.smartprogramming)

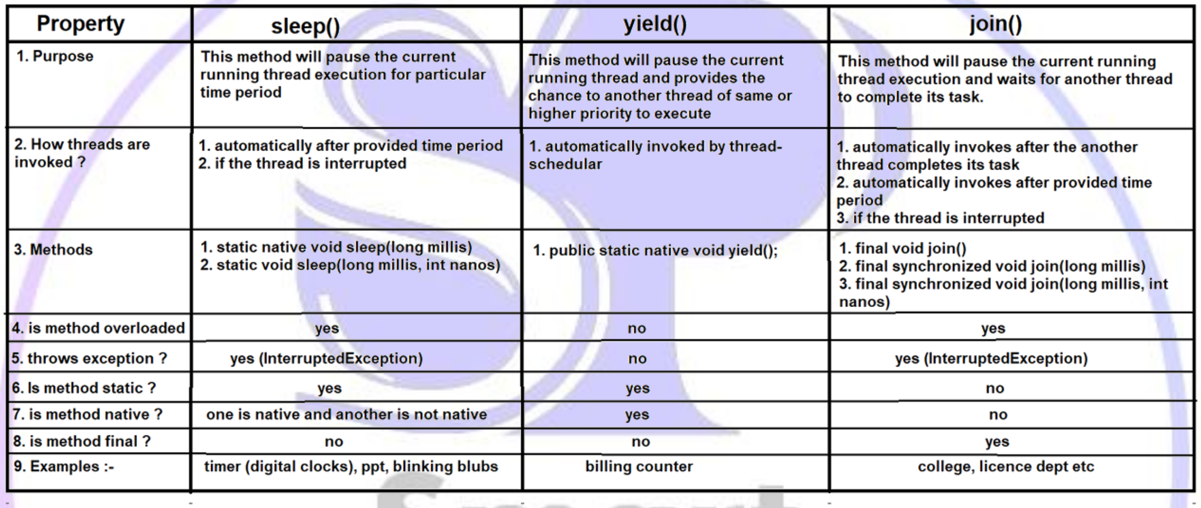
https://www.facebook.com/smartprogramming.india

https://www.instagram.com/smart_programming

Java
android
python
Data Science
Machine Learning
Artificial Intelligence
M E A N
Full Stack
BIG DATA
hadoop

Thread class methods (Part 3) in Java

=> What is difference between sleep(), yield() and join()



Property	sleep()	yield()	join()
1. Purpose	This method will pause the current running thread execution for particular time period	This method will pause the current running thread and provides the chance to another thread of same or higher priority to execute	This method will pause the current running thread execution and waits for another thread to complete its task.
2. How threads are invoked ?	1. automatically after provided time period 2. if the thread is interrupted	1. automatically invoked by thread-schedular	1. automatically invokes after the another thread completes its task 2. automatically invokes after provided time period 3. if the thread is interrupted
3. Methods	1. static native void sleep(long millis) 2. static void sleep(long millis, int nanos)	1. public static native void yield();	1. final void join() 2. final synchronized void join(long millis) 3. final synchronized void join(long millis, int nanos)
4. is method overloaded	yes	no	yes
5. throws exception ?	yes (InterruptedException)	no	yes (InterruptedException)
6. Is method static ?	yes	yes	no
7. is method native ?	one is native and another is not native	yes	no
8. is method final ?	no	no	yes
9. Examples :-	timer (digital clocks), ppt, blinking blubs	billing counter	college, licence dept etc

---

=> Thread class methods :-

6. Thread interrupting methods :-

- ➔ interrupt() - It is used to interrupt the sleeping or waiting thread
- ➔ isInterrupted() - It is used to check the thread interrupt status. It will return true if interrupt status is true otherwise it will return false

- ➔ interrupted() - It is used to check the thread interrupt status but if the interrupt status is true it will change the interrupt status into false.

=> interrupt() :-

- ➔ It is used to interrupt the thread
- ➔ It will work only when the thread is in sleeping or waiting state otherwise interrupt statement will be of no use
- ➔ It throws an exception "java.lang.InterruptedException"

=> What is difference between isInterrupted() and interrupted() method ?

1. isInterrupted() method does not change the interrupt status
   interrupted() method change the interrupt status

2. isInterrupted() method is non-static method
   interrupted() method is static method

Company Links & Contacts

Company Name: Smart Programming (+91 62838-30308)

Address : Chandigarh & Mohali (Punjab), India

Websites: https://www.smartprogramming.in/

https://courses.smartprogramming.in

Android App:
https://play.google.com/store/apps/details?id=com.smartprogramming

YouTube Channel:
https://www.youtube.com/c/SmartProgramming

Smart Programming : YouTube Channel

An investment in Knowledge pays the best interest….

Smart Programming

We Educate - We Develop

☎ +91 62838-30308

Call us to Learn Latest Technologies

City : Mohali (Punjab), & Chandigarh (India)

WEBSITE : http://www.smartprogramming.in

BUY COURSES ON : https://courses.smartprogramming.in

YOUTUBE CHANNEL : Smart Programming (https://www.youtube.com/c/SmartProgramming)

ANDROID APP : Smart Programming (https://play.google.com/store/apps/details?id=com.smartprogramming)

https://www.facebook.com/smartprogramming.india

https://www.instagram.com/smart_programming

Java
android
python
Data Science
Machine Learning
Artificial Intelligence
M E A N
Full Stack
BIG DATA
hadoop

Synchronization in Java (Part 1)

=> NOTE : Whenever we execute multiple threads, all the threads starts their execution and thus the output is not constant. And due to this data inconsistency problem may occur. This problem can occur in case of projects like banking, reservation (bus, train, airoplane etc), theater booking etc

- ⇨ To solve the above problem we use synchronization concept

---

=> What is synchronization ?

- ➔ Synchronization is the process by which we can control the multiple threads execution for a particular shared resource

=> Problems without synchronization :-

- ➔ Data Inconsistancy
- ➔ Thread Interference

=> Advantage of synchronization :-

- ➔ Resolves the Data Inconsistancy problem
- ➔ Resolves the Thread Interference problem

=> Types of synchronization :-

1. Process Synchronization (not the part of multithreading)
2. Thread Synchronization
    - 2.1 Mutual Exclusive
        - = Can be achieved by 3 ways
            - -> By "synchronized method"
            - -> By "synchronized block"
            - -> By "static synchronization"

2.2 Cooperation (inter-thread communication)

= Can be achieved by methods

-> wait(), notify() & notifyAll() methods

=> How to achieve synchronization :-

➔ We can achieve synchronization by 4 ways :-

1. By "synchronized method"
2. By "synchronized block"
3. By "static synchronization"

4. By using wait(), notify() and notifyAll() methods

=> "synchronized method" :-

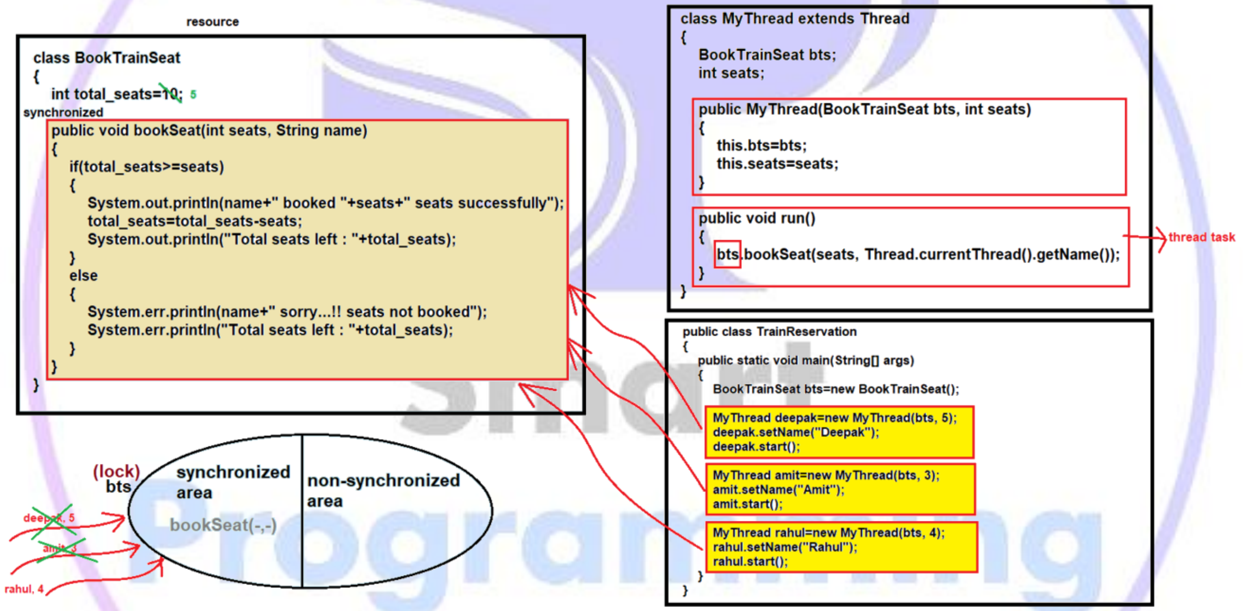
- ➔ If we declare any method as synchronized by "synchronized" keyword, that method is known as "synchronized method"
- ➔ Syntax :

```
synchronized void show()
{


}
```

=> Internal working of synchronization :-

- ➔ In case of synchronization object lock comes in the action. Every object has one unique lock. Whenever any thread tries to enter the synchronized area that thread has to acquire the object lock. After acquiring the lock that thread can access the synchronized area and all other thread has to wait to access that synchronized area.

After the thread (the thread who has acquired the lock) execution completes, it will release the lock and other thread can again acquire the lock and can enter the synchronized area.



```
class BookTrainSeat
{
   int total_seats=10;
synchronized
   public void bookSeat(int seats, String name)
   {
      if(total_seats>=seats)
      {
         System.out.println(name+" booked "+seats+" seats successfully");
         total_seats=total_seats-seats;
         System.out.println("Total seats left : "+total_seats);
      }
      else
      {
         System.err.println(name+" sorry...!! seats not booked");
         System.err.println("Total seats left : "+total_seats);
      }
   }
}
```

```
class MyThread extends Thread
{
   BookTrainSeat bts;
   int seats;

   public MyThread(BookTrainSeat bts, int seats)
   {
      this.bts=bts;
      this.seats=seats;
   }

   public void run()
   {
      bts.bookSeat(seats, Thread.currentThread().getName());
   }
}
```

```
public class TrainReservation
{
   public static void main(String[] args)
   {
      BookTrainSeat bts=new BookTrainSeat();

      MyThread deepak=new MyThread(bts, 5);
      deepak.setName("Deepak");
      deepak.start();

      MyThread amit=new MyThread(bts, 3);
      amit.setName("Amit");
      amit.start();

      MyThread rahul=new MyThread(bts, 4);
      rahul.setName("Rahul");
      rahul.start();
   }
}
```

=> "synchronized block"

- ➔ Syntax :-

```
synchronized(object reference)
{


}
```

- ➔ "synchronized block" is used to achieve synchronization but on the specific lines of code
- ➔ The scope of "synchronized block" is smaller than the "synchronized method"
- ➔ It is always preferable to use "synchronized block" in order to increase the application performance

Company Links & Contacts

Company Name: Smart Programming (+91 62838-30308)

Address : Chandigarh & Mohali (Punjab), India

Websites: https://www.smartprogramming.in/

https://courses.smartprogramming.in

Android App:
https://play.google.com/store/apps/details?id=com.smartprogramming

YouTube Channel:
https://www.youtube.com/c/SmartProgramming

Smart Programming : YouTube Channel

An investment in Knowledge pays the best interest….

Smart Programming
We Educate - We Develop

+91 62838-30308
Call us to Learn
Latest Technologies

City : Mohali (Punjab),
& Chandigarh
(India)

WEBSITE : http://www.smartprogramming.in

BUY COURSES ON : https://courses.smartprogramming.in

YOUTUBE CHANNEL : Smart Programming (https://www.youtube.com/c/SmartProgramming)

ANDROID APP : Smart Programming
(https://play.google.com/store/apps/details?id=com.smartprogramming)

https://www.facebook.com/smartprogramming.india

https://www.instagram.com/smart_programming

Java
android
python
Data Science
Machine Learning
Artificial Intelligence
M E A N
Full Stack
BIG DATA
hadoop

Synchronization in Java (Part 2)

=> "static synchronization" :-

- ➔ In this part, lock comes for class level. All other working is same.

=> wait(), notify() and notifyAll() methods :-

- ➔ These methods are used for inter-thread communication
- ➔ These methods are of Object class
- ➔ To call these methods on any object, thread should own the lock of that object i.e. the thread should be inside the synchronized area.

- ➔ **wait() method :-**
  - o This method will release the current running thread lock and waits until it will get the notification

- **notify() method :-**
  - This method is used to wake up the single waiting thread and releases the object lock.
- **notifyAll() method :-**
  - This method is used to wake up all the waiting thread and releases the object lock.

=> What is difference between sleep() and wait() method :-

1. sleep() is the Thread class method

   wait() is the Object class method

2. sleep() is static method

   wait() is non-static method

3. sleep() method doesnt releases the lock

wait() method releases the lock

4. sleep() method is invoked after the provided time period

wait() method can be invoked by notify() or notifyAll() or by provided time period

---

Company Links & Contacts

Company Name: Smart Programming (+91 62838-30308)

Address : Chandigarh & Mohali (Punjab), India

Websites: https://www.smartprogramming.in/

https://courses.smartprogramming.in

Android App:
https://play.google.com/store/apps/details?id=com.smartprogramming

YouTube Channel:
https://www.youtube.com/c/SmartProgramming

Smart Programming : YouTube Channel

An investment in Knowledge pays the best interest….

Smart Programming

We Educate - We Develop

☎ +91 62838-30308

Call us to Learn Latest Technologies

City : Mohali (Punjab), & Chandigarh (India)

WEBSITE : http://www.smartprogramming.in

BUY COURSES ON : https://courses.smartprogramming.in

YOUTUBE CHANNEL : Smart Programming (https://www.youtube.com/c/SmartProgramming)

ANDROID APP : Smart Programming (https://play.google.com/store/apps/details?id=com.smartprogramming)

https://www.facebook.com/smartprogramming.india

https://www.instagram.com/smart_programming

Java
android
python
Data Science
Machine Learning
Artificial Intelligence
M E A N
Full Stack
BIG DATA
hadoop

Synchronization in Java (Part 3)

=> Disadvantages of Synchronization :-

1. It will decrease the application performance because the threads has to wait for the object lock
2. Due to synchronization "deadlock" condition may occur

=> Deadlock :-

- ➔ Deadlock is the situation where one thread is waiting for an object lock that is acquired by another thread and second thread is waiting for an object lock which is acquired by first thread. In this type of situation both threads will wait for each other locks to release and this situation goes for infinite time, thus deadlock condition will occur

=> Problems with "synchronized" keyword :-

1. If we want to acquire the lock in one method and release the lock in another method then this cannot be dont with synchronized keyword
2. There is no flexibility to try for the lock without waiting
3. There is no way to get the list of all the waiting threads
4. We cannot judge/control which thread will acquire the lock when the first thread will release the lock

=> To overcome these problems java introduced java.util.concurrent.locks package (Lock interface & ReentrantLock class) in JDK 1.5 version

=> Lock interface :-

- ➔ Lock interface is used to achieve synchronization same as synchronized blocks
- ➔ Syntax :

```
        public interface Lock
        {
                void lock();
                void unlock();
                boolean tryLock();
                boolean tryLock(-) throws
InterruptedException;
        }
```

=> ReentrantLock class :-

- ➔ It is the implemented class of Lock interface.
- ➔ Syntax :-

public class ReentrantLock implements Lock, Serializable

{

//constructors :-

1. public ReentrantLock();

2. public ReentrantLock(boolean fair);

// contains all the methods of Lock interface

final int getHoldCount();

final boolean isLocked();

public boolean isHeldByCurrentThread();

```
        public final int getQueueLength();
    }
```

=> Note :-

1. We should always use unlock() method in finally block
2. If we use lock() method multiple times then we have to unlock it multiple times only

---

Company Links & Contacts

Company Name: Smart Programming (+91 62838-30308)

Address : Chandigarh & Mohali (Punjab), India

Websites: https://www.smartprogramming.in/

https://courses.smartprogramming.in

Android App: https://play.google.com/store/apps/details?id=com.smartprogramming

YouTube Channel: https://www.youtube.com/c/SmartProgramming

Smart Programming : YouTube Channel

An investment in Knowledge pays the best interest….

Smart Programming

We Educate - We Develop

☎ +91 62838-30308

Call us to Learn Latest Technologies

City : Mohali (Punjab), & Chandigarh (India)

WEBSITE : http://www.smartprogramming.in

BUY COURSES ON : https://courses.smartprogramming.in

YOUTUBE CHANNEL : Smart Programming (https://www.youtube.com/c/SmartProgramming)

ANDROID APP : Smart Programming (https://play.google.com/store/apps/details?id=com.smartprogramming)

https://www.facebook.com/smartprogramming.india

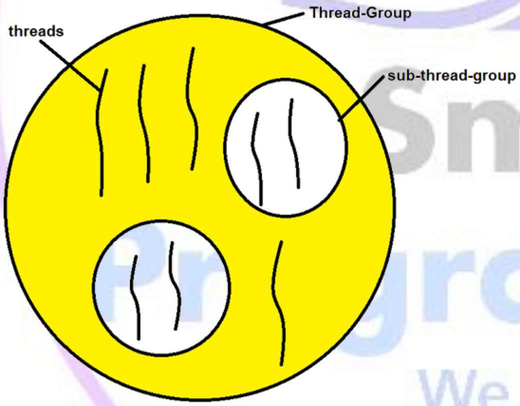
https://www.instagram.com/smart_programming

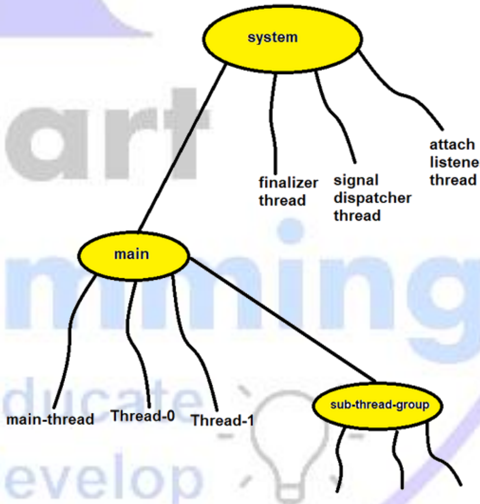
Java
android
python
Data Science
Machine Learning
Artificial Intelligence
M E A N
Full Stack
BIG DATA
hadoop

ThreadGroup class & Thread-Pool Framework

=> ThreadGroup :-

- ➔ It is the group of several threads into a single unit or object
- ➔ A thread-group can contain multiple thread-group
- ➔ The thread-group creates a tree in which every thread-group will a single parent thread-group


threads
Thread-Group
sub-thread-group


system
finalizer thread
signal dispatcher thread
attach listener thread
main
main-thread
Thread-0
Thread-1
sub-thread-group

- ➔ Every thread-group belongs to some thread-group except system thread-group
- ➔ "system" thread-group is the root group of all the thread-groups in java

- ➔ Main thread belongs to the "main" thread-group
- ➔ The main advantage of thread-group is we can perform several common operations very easily in order to improve our application performance
- ➔ For example : creating groups in messenger or mails and sending message

- ➔ ThreadGroup is the class which is present in java.lang package

- ➔ Constructors of ThreadGroup :-

1. public ThreadGroup(String name);
2. public ThreadGroup(ThreadGroup groupname, String name)

- ➔ Methods of ThreadGroup :-

1. activeCount()
2. activeGroupCount()
3. getMaxPriority()
4. getName()
5. getParent()

6. setMaxPriority()
7. list()
8. isDaemon()
9. setDaemon()
10. interrupt()
11. suspend()
12. destroy()

➔ A current running thread can only get the information about current thread-group but not the parent/child thread-group

---

=> ThreadPool :-

➔ ThreadPool represents a group of worker threads that are waiting for the job and can be reused many times

- ➔ "ThreadPool framework" is also known as "Executor framework"
- ➔ ThreadPool was introduced in JDK 1.5 version

- ➔ Advantage : It saves a lot of time in creating threads and providing jobs to them which leads to the better performance of our application

- ➔ ThreadPool is already implemented in Servlet & JSP containers which each request is processed by thread which comes from thread-pool

- ➔ ThreadPool framework contains following classes and interfaces :-

1. Executors (class) (important)
2. ExecutorService (interface) (important)
3. Executor (interface)
4. ExecutorCompletionService (class)

=> Executors class :-

- ➔ Executors class provides factory and utility methods for ExecutorService, Executor, ThreadFactory, ScheduledExecutor Service & Callable.
- ➔ Methods of Executors class :

1. Executors.newFixedThreadPool(int no_of_threads)
2. Executors.newSingleThreadExecutor()
3. Executors.newCachedThreadPool()
4. Executors.newScheduledThreadPool()
5. Executors.newSingleThreadScheduledExecutor()

=> ExecutorService interface :-

- ➔ ExecutorService interface allows us to execute the thread task asynchronously.
- ➔ ExecutorService helps in maintaining a pool of threads and assign them tasks. It also provides the facility to queue up the tasks until there is any free thread available.

➔ ExecutorService defines the methods that executes the threads and returns some results.

➔ Methods of ExecutorService interface :-

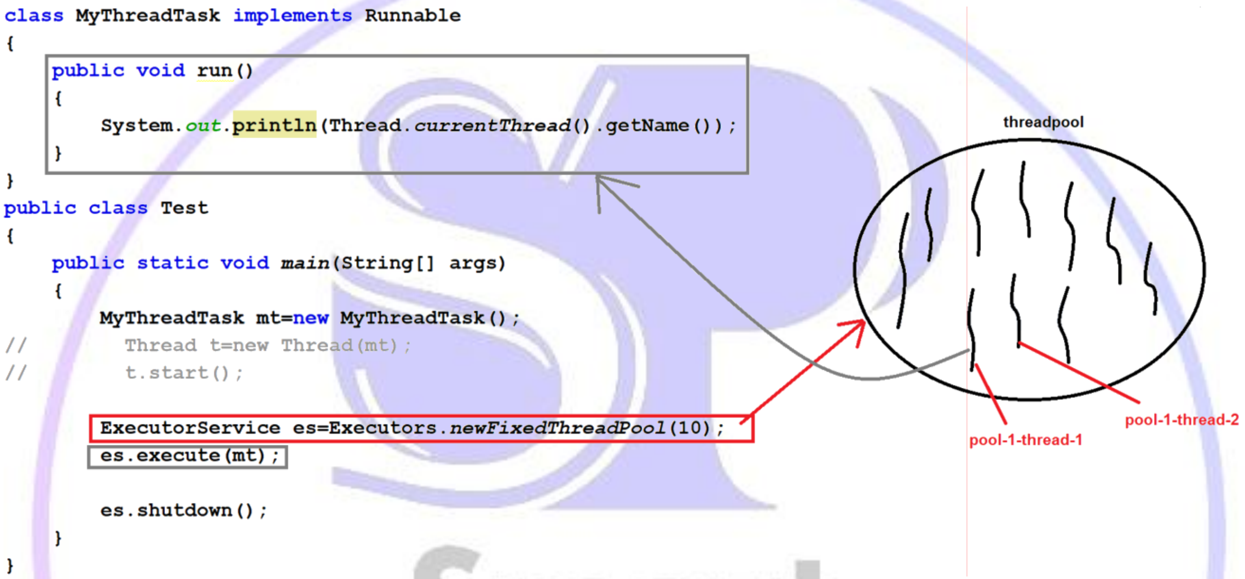
- Below methods will assign the task to ExecutorService :-

1. execute(Runnable r)
2. submit(Runnable/Callable r)
3. invokeAny(Collection c)
4. invokeAll(Collection c)

➔ Below method is used for shutdown the ExecutorService :-

5. shutdown()
6. shutdownNow()

=> Program :-



```
class MyThreadTask implements Runnable
{
    public void run()
    {
        System.out.println(Thread.currentThread().getName());
    }
}
public class Test
{
    public static void main(String[] args)
    {
        MyThreadTask mt=new MyThreadTask();
//        Thread t=new Thread(mt);
//        t.start();

        ExecutorService es=Executors.newFixedThreadPool(10);
        es.execute(mt);

        es.shutdown();
    }
}
```

=> ThreadPool have some risks which are as follows :-

1. Deadlock
2. Thread interference
3. Thread leakage : This risk can occur if any thread removed from the thread-pool but that thread didnt returned in the thread-pool
4. Resource thrashing : This risk can occur when there are lot of thread in the thread-pool, then time will be wasted in context-switching between the threads.

Company Links & Contacts

Company Name: Smart Programming (+91 62838-30308)

Address : Chandigarh & Mohali (Punjab), India

Websites: https://www.smartprogramming.in/

https://courses.smartprogramming.in

Android App:
https://play.google.com/store/apps/details?id=com.smartprogramming

YouTube Channel:
https://www.youtube.com/c/SmartProgramming

Smart Programming : YouTube Channel

An investment in Knowledge pays the best interest….

Smart Programming
We Educate - We Develop

+91 62838-30308
Call us to Learn
Latest Technologies

City : Mohali (Punjab),
& Chandigarh
(India)

WEBSITE : http://www.smartprogramming.in

BUY COURSES ON : https://courses.smartprogramming.in

YOUTUBE CHANNEL : Smart Programming (https://www.youtube.com/c/SmartProgramming)

ANDROID APP : Smart Programming
(https://play.google.com/store/apps/details?id=com.smartprogramming)

https://www.facebook.com/smartprogramming.india

https://www.instagram.com/smart_programming

Java
android
python
Data Science
Machine Learning
Artificial Intelligence
M E A N
Full Stack
BIG DATA
hadoop

Callable & ThreadLocal class

=> Callable & Future :-

- ➔ We have total 3 ways to create threads in java

1. By extending "Thread" class
2. By implementing "Runnable" interface
3. By implementing "Callable" interface

=> Callable :-

- ➔ The main advantage to use Callable interface is that it returns some value after the thread completes its task
- ➔ Callable is an interface which is present in java.util.concurrent package.
- ➔ Callable contains only one method i.e. Object call() throws Exception
- ➔ Callable is a FunctionalInterface because it contains only one method

=> Future :-

- ➔ Future is an interface which is present in java.util.concurrent package
- ➔ A Future represents the result of an asynchronous computation
- ➔ In simple way we can say that Future is used to store the returned value of thread

- ➔ Methods of Future interface :-

1. V get() throws InterruptedException, ExecutionException;
2. V get(long timeout, TimeUnit unit) throws InterruptedException, ExecutionException, TimeoutException;
3. boolean cancel(boolean mayInterruptIfRunning);
4. boolean isCancelled();
5. boolean isDone();

=> What is difference between Runnable & Callable ?

1. We should use Runnable interface when our thread does not return any value
   We should use Callable interface when our thread returns some value after execution completes

2. Runnable interface is present in java.lang package
   Callable interface is present in java.util.concurrent package

3. Runnable interface was introduced in JDK 1.0 version
   Callable interface was introduced in JDK 1.5 version

4. Runnable interface has one method i.e. run() method
   Callable interface has one method i.e. call() method

5. Runnable method (run() method) does not throw any exception
   Callable method (call() method) throws an exception

=> ThreadLocal :-

➔ ThreadLocal provides the thread-local variables. It means that we can create a variable in reference to the thread which can be manipulated or removed by that thread only

- ➔ Examples are we can pass the user id or transaction id for users request

- ➔ ThreadLocal is the class which is present in java.lang package
- ➔ ThreadLocal concept was introduced in JDK 1.5 version
- ➔ ThreadLocal methods :

1. get()
2. set(Object obj)
3. remove()

=> Note :-

1. ThreadLocal concept was introduced in JDK 1.2 version and was enhanced in JDK 1.5 verison
2. A thread can access its own thread-local variable and it cannot access any other thread-local variable value

3. If the thread does to dead state then that thread-local variable is eligible for garbage collection

Company Links & Contacts

Company Name: Smart Programming (+91 62838-30308)

Address : Chandigarh & Mohali (Punjab), India

Websites: https://www.smartprogramming.in/

https://courses.smartprogramming.in

Android App: https://play.google.com/store/apps/details?id=com.smartprogramming

YouTube Channel: https://www.youtube.com/c/SmartProgramming

Smart Programming : YouTube Channel

An investment in Knowledge pays the best interest….

Smart Programming
We Educate - We Develop

☎ +91 62838-30308
Call us to Learn
Latest Technologies

City : Mohali (Punjab), & Chandigarh (India)

WEBSITE : http://www.smartprogramming.in

BUY COURSES ON : https://courses.smartprogramming.in

YOUTUBE CHANNEL : Smart Programming (https://www.youtube.com/c/SmartProgramming)

ANDROID APP : Smart Programming (https://play.google.com/store/apps/details?id=com.smartprogramming)

https://www.facebook.com/smartprogramming.india

https://www.instagram.com/smart_programming

Java
android
python
Data Science
Machine Learning
Artificial Intelligence
M E A N
Full Stack
BIG DATA
hadoop

Byte Stream in Java

Java IO :-

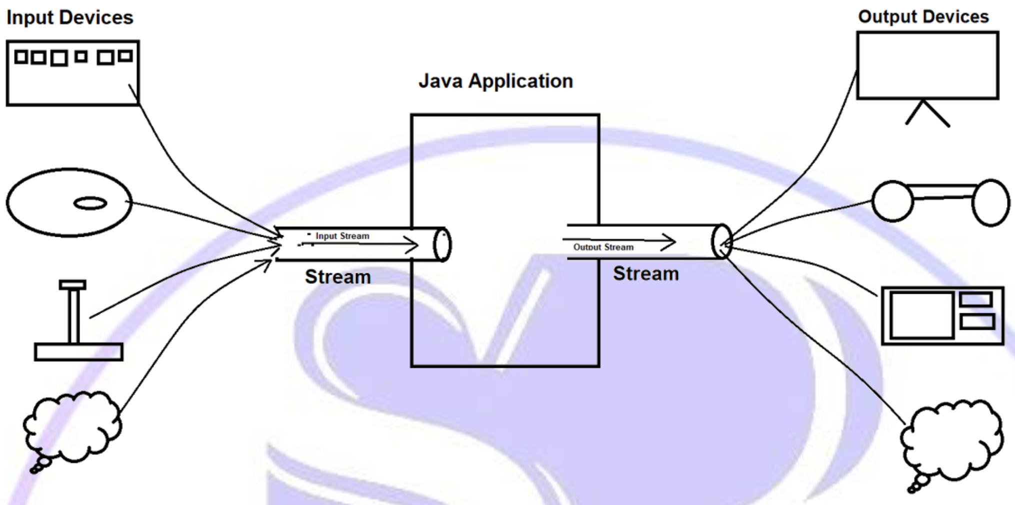
Terms used in Java IO :-

1. Input Devices : Keyboard, mouse, Optical pen, Joystick etc

2. Output Devices : Screen, Head Set, Printers, Speakers etc

3. Stream : Flow of data, Real world example Water flow

=> Stream : Stream is a medium which allow the data to flow from input devices to java application or from java application to output devices


Input Devices
Java Application
Output Devices
Input Stream
Stream
Output Stream
Stream

-> In java there are 3 predefined stream which transfer the data from CMD to java application which are as follows :-

1. System.out
2. System.in
3. System.err

-> To transfer the data from input devices to java application or java application to output devices, java has provided many classes which are present in java.io package

-> Types of Streams :-

1. Byte Stream
2. Character Stream

=> Byte Stream :-

-> In this stream data is transfered in the form of bytes i.e. 0 and 1

-> The length of byte stream is 1 byte

-> Types of byte stream :-

1. InputStream

-> public int read(-) throws IOException

-> public int available() throws IOException

-> public void close() throws IOException

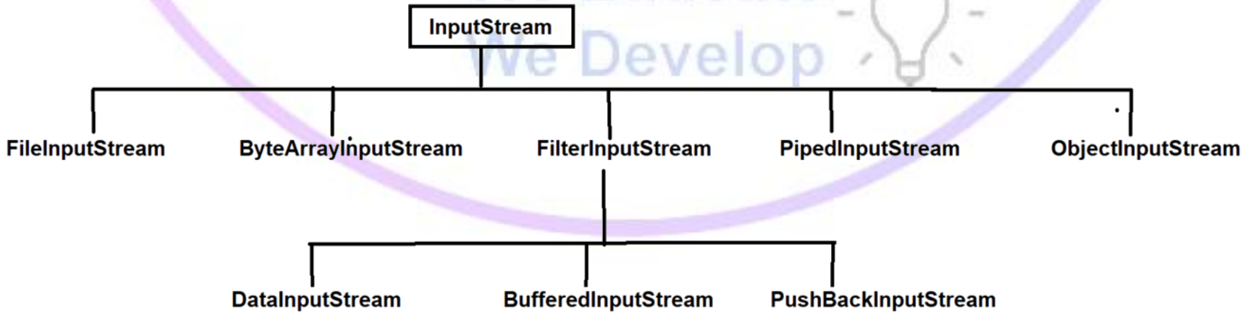
2. OutputStream

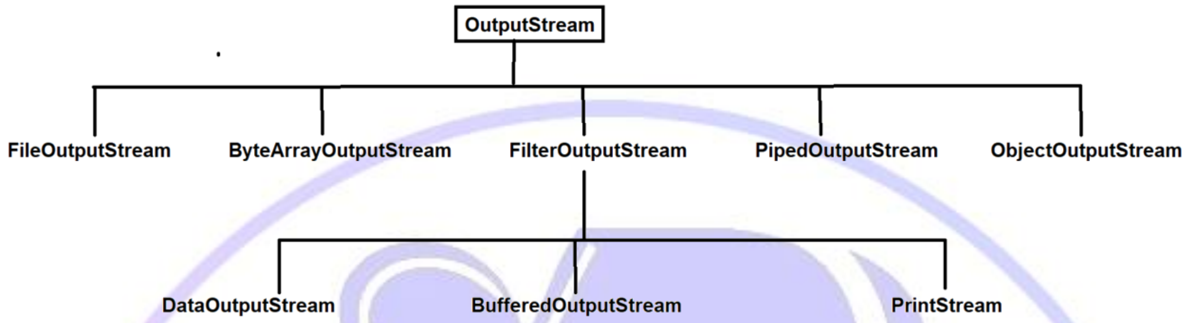
-> public void write(-) throws IOException

-> public void flush() throws IOException

-> public void close() throws IOException

-> Hierarchy of InputStream and OutputStream classes :-


InputStream
FileInputStream
ByteArrayInputStream
FilterInputStream
PipedInputStream
ObjectInputStream
DataInputStream
BufferedInputStream
PushBackInputStream


OutputStream
FileOutputStream
ByteArrayOutputStream
FilterOutputStream
PipedOutputStream
ObjectOutputStream
DataOutputStream
BufferedOutputStream
PrintStream

Company Links & Contacts

Company Name: Smart Programming (+91 62838-30308)

Address : Chandigarh & Mohali (Punjab), India

Websites: https://www.smartprogramming.in/
https://courses.smartprogramming.in

Android App:
https://play.google.com/store/apps/details?id=com.smartprogramming

YouTube Channel:
https://www.youtube.com/c/SmartProgramming

Smart Programming : YouTube Channel

An investment in Knowledge pays the best interest….

Smart Programming
We Educate - We Develop

☎ +91 62838-30308
Call us to Learn Latest Technologies

City : Mohali (Punjab), & Chandigarh (India)

WEBSITE : http://www.smartprogramming.in

BUY COURSES ON : https://courses.smartprogramming.in

YOUTUBE CHANNEL : Smart Programming (https://www.youtube.com/c/SmartProgramming)

ANDROID APP : Smart Programming (https://play.google.com/store/apps/details?id=com.smartprogramming)

https://www.facebook.com/smartprogramming.india

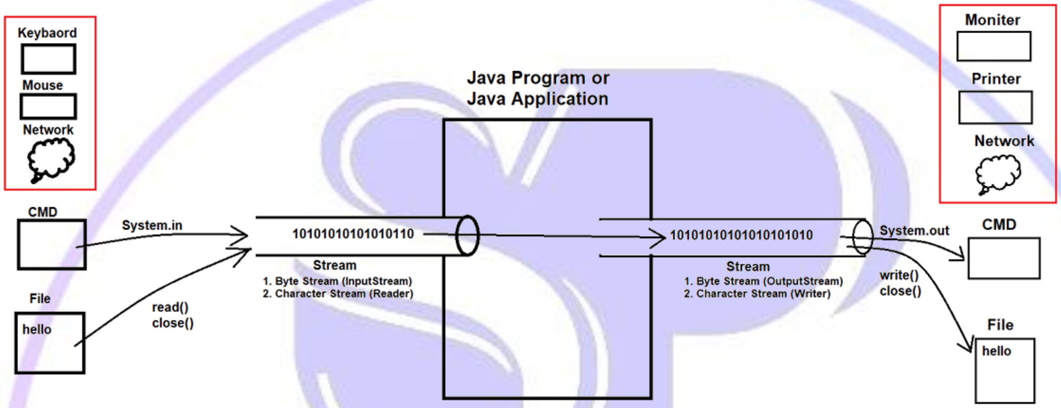
https://www.instagram.com/smart_programming

Java
android
python
Data Science
Machine Learning
Artificial Intelligence
M E A N
Full Stack
BIG DATA
hadoop

Character Stream in Java

& Dynamic Input Approches

InputStream & OutputStream Diagram :-


Keybaord
Mouse
Network
CMD
File
hello
System.in
read()
close()
10101010101010110
Stream
1. Byte Stream (InputStream)
2. Character Stream (Reader)
Java Program or
Java Application
1010101010101010101010
Stream
1. Byte Stream (OutputStream)
2. Character Stream (Writer)
System.out
write()
close()
Moniter
Printer
Network
CMD
File
hello

=> Character Stream :

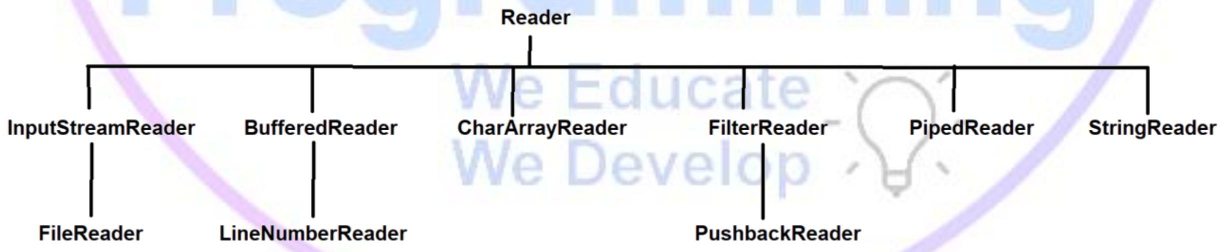
-> It is used to transfer the data in characters form

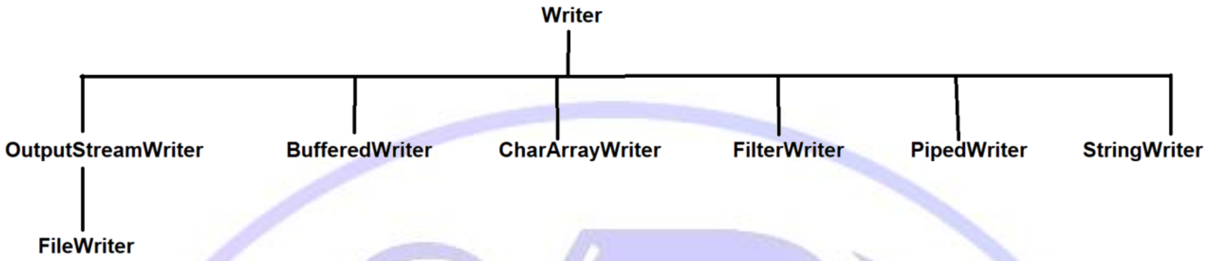
-> In character stream the length of data is 2 bytes

-> Types of Character Stream :-

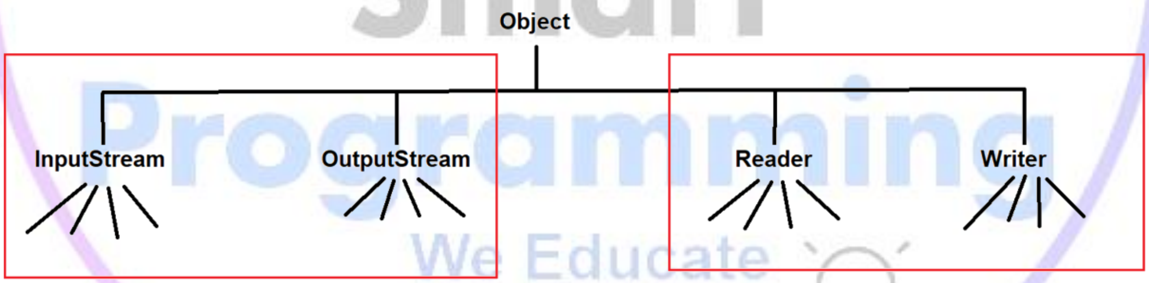
1. Reader
    - -> read(-)
    - -> close()
2. Writer
    - -> write()
    - -> close()

-> Hierarchy of Writer & Reader classes


Reader
InputStreamReader
BufferedReader
CharArrayReader
FilterReader
PipedReader
StringReader
FileReader
LineNumberReader
PushbackReader


Writer
OutputStreamWriter
BufferedWriter
CharArrayWriter
FilterWriter
PipedWriter
StringWriter
FileWriter

Object is the parent class or all the classes in java and thus InputStream, OutputStream, Reader & Writer classes comes directly in Object class

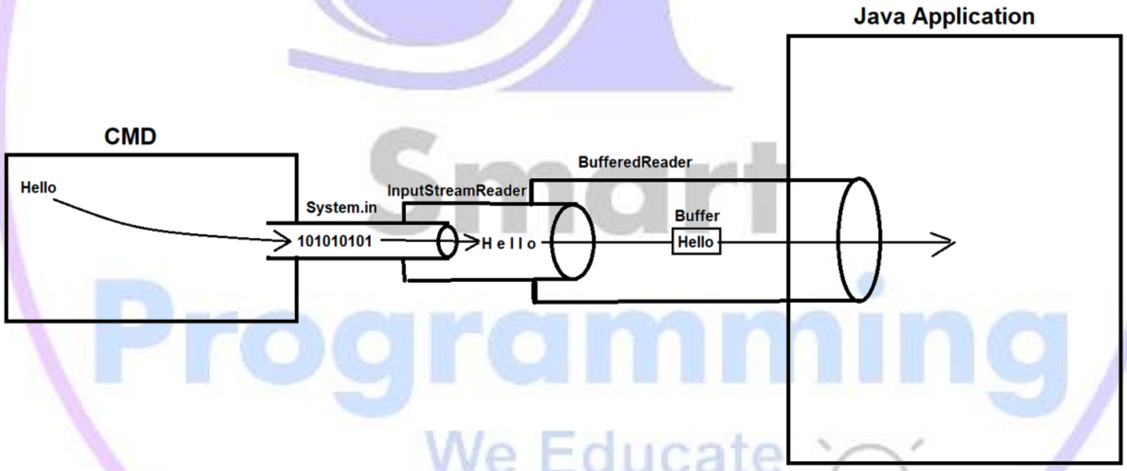

Object
InputStream
OutputStream
Reader
Writer

=> Dynamic Input Approches :-

-> Dynamic Input Approch is the way by which programmer take the input value at runtime

-> There are 3 ways for dynamic input approch

1. Scanner
2. BufferedReader


Java Application
CMD
Hello
System.in
101010101
InputStreamReader
H e l l o
BufferedReader
Buffer
Hello

3. Console

Company Links & Contacts

Company Name: Smart Programming (+91 62838-30308)

Address : Chandigarh & Mohali (Punjab), India

Websites: https://www.smartprogramming.in/

https://courses.smartprogramming.in

Android App:
https://play.google.com/store/apps/details?id=com.smartprogramming

YouTube Channel:
https://www.youtube.com/c/SmartProgramming

Smart Programming : YouTube Channel

An investment in Knowledge pays the best interest….

Smart Programming
We Educate - We Develop

☎ +91 62838-30308
Call us to Learn Latest Technologies
City : Mohali (Punjab), & Chandigarh (India)

WEBSITE : http://www.smartprogramming.in

BUY COURSES ON : https://courses.smartprogramming.in

YOUTUBE CHANNEL : Smart Programming (https://www.youtube.com/c/SmartProgramming)

ANDROID APP : Smart Programming (https://play.google.com/store/apps/details?id=com.smartprogramming)

https://www.facebook.com/smartprogramming.india

https://www.instagram.com/smart_programming

Java
android
python
Data Science
Machine Learning
Artificial Intelligence
M E A N
Full Stack
BIG DATA
hadoop

InputStream & OutputStream classes in Java

InputStream & OutputStream Diagram :-

=> FileInputStream :-

-> FileInputStream is used to read the data for eg text, image, video, audio etc in the form of bytes

-> FileInputStream can read the character stream data also, but you should use FileReader to read character type of data

-> In this case, if file/data is not found, then it will throw FileNotFoundException

=> FileOutputStream :-

-> FileOutputStream is used to write the data for eg text, image, video, audio etc in the form of bytes

-> FileOutputStream can read the character stream data also but you should use FileWriter to write character stream data

-> In this case if file is not found, then it will create new file

=> ByteArrayInputStream :-

-> ByteArray + InputStream : As name suggests, ByteArrayInputStream can be used to read ByteArray data only as an input stream

-> ByteArrayInputStream uses internal buffer to read the byte array data

=> ByteArrayOutputStream :-

-> ByteArrayOutputStream is used to write the common data to multiple files

=> FilterInputStream :

-> FileInputStream is less used because it provides sub-classes for more functionalities i.e. DataInputStream, BufferedInputStream & PushBackInputStream

=> FilterOutputStream :-

-> FilterOutputStream is less used because it provides sub-class for more functionalities i.e. DataOutputStream, BufferedOutputStream & PrintStream

---

=> DataInputStream :-

-> DataInputStream is used to read the primitive data from input stream in a machine-independent way

=> DataOutputStream :-

-> DataOutputStream is used to write the primitive data form input stream in a machine-independent way

---

=> BufferedInputStream :-

-> In this case buffer machanish is used internally which will improve the performance

=> BufferedOutputStream :-

-> BufferedOutputStream also use buffer to write the data in order to improve the performance

(flush() method is used forcefully write the data into the stream or output before erasing the buffer memory)

=> PushBackInputStream :

-> PushBackInputStream can be used to unread a byte which is already read and push back one byte

---

=> PrintStream :-

-> PrintStream provides the methods to write the data into another stream

-> PrintStream class automatically flushes the data so there is no need to use flush() method

-> PrintStream method does not throw IOException

=> PipedInputStream & PipedOutputStream :-

-> PipedInputStream and PipedOutputStream are used simultaneously

-> These streams are used to transfer the data from one thread to another thread

-> Both these streams should be connected

=> ObjectInputStream & ObjectOutputStream :-

-> These streams are used in serialization

Company Links & Contacts

Company Name: Smart Programming (+91 62838-30308)

Address : Chandigarh & Mohali (Punjab), India

Websites: https://www.smartprogramming.in/

https://courses.smartprogramming.in

Android App: https://play.google.com/store/apps/details?id=com.smartprogramming

YouTube Channel: https://www.youtube.com/c/SmartProgramming

Smart Programming : YouTube Channel

An investment in Knowledge pays the best interest....

Smart Programming
We Educate - We Develop

☎ +91 62838-30308
Call us to Learn
Latest Technologies

City : Mohali (Punjab),
& Chandigarh
(India)

WEBSITE : http://www.smartprogramming.in

BUY COURSES ON : https://courses.smartprogramming.in

YOUTUBE CHANNEL : Smart Programming (https://www.youtube.com/c/SmartProgramming)

ANDROID APP : Smart Programming (https://play.google.com/store/apps/details?id=com.smartprogramming)

https://www.facebook.com/smartprogramming.india

https://www.instagram.com/smart_programming

Java
android
python
Data Science
Machine Learning
Artificial Intelligence
M E A N
Full Stack
BIG DATA
hadoop

Reader & Writer classes explanation in Java

=> Reader & Writer Classes Explanation :-

=> InputStreamReader :-

-> InputStreamReader is a bridge which take the input from byte stream and decodes into character stream

=> OutputStreamWriter :-

-> OutputStreamWriter converts the character stream into byte stream using a specified charset

---

=> FileReader :-

-> It inherits the InputStreamReader class

-> FileReader is used to read character stream

=> FileWriter :-

-> FileWriter class inherits OutputStreamWriter class

-> FileWriter is used to write character stream

=> BufferedReader :-

-> BufferedReader is used to read the text from character stream

-> BufferedReader can be used to read data line by line by readLine() method

-> Performance of BufferedReader is fast

=> BufferedWriter :-

-> BufferedWriter is used to write the char array or string into the file

-> BufferedWriter performance is fast

=> LineNumberReader :-

-> LineNumberReader is a buffered character input stream which keeps the track of line numbers

---

=> CharArrayReader :-

-> CharArray + Reader : As the name suggests it is used to read the character array stream

=> CharArrayWriter :-

-> CharArray + Writer : It is used to write the common data to multiple file

---

=> FilterReader :-

-> FilterReader is used to perform filtering operations on reader stream

-> FilterReader is an abstract class

=> FilterWriter :-

-> FilterWriter class is used to write filtered character stream

-> FilterWriter is an abstract class

=> PushbackReader :-

-> PushbackReader is used to push back the one character into stream and read again

---

=> PipedReader :-

-> PipedReader is used to read the content from a pipe as character stream

-> It is used in case of multiple threads

=> PipedWriter :-

-> PipedWriter is used to write the content to a pipe as a character stream

-> It is also used in case of multiple threads

=> StringReader :-

-> StringReader is character stream which string as a source

=> StringWriter :-

-> StringWriter is a character stream which can be used to write data from string as a source

---

Company Links & Contacts

Company Name: Smart Programming (+91 62838-30308)

Address : Chandigarh & Mohali (Punjab), India

Websites: https://www.smartprogramming.in/

https://courses.smartprogramming.in

Android App:
https://play.google.com/store/apps/details?id=com.smartprogramming

YouTube Channel:
https://www.youtube.com/c/SmartProgramming

Smart Programming : YouTube Channel

An investment in Knowledge pays the best interest….

Smart Programming
We Educate - We Develop

☎ +91 62838-30308
Call us to Learn
Latest Technologies

City : Mohali (Punjab),
& Chandigarh
(India)

WEBSITE : http://www.smartprogramming.in

BUY COURSES ON : https://courses.smartprogramming.in

YOUTUBE CHANNEL : Smart Programming (https://www.youtube.com/c/SmartProgramming)

ANDROID APP : Smart Programming
(https://play.google.com/store/apps/details?id=com.smartprogramming)

https://www.facebook.com/smartprogramming.india

https://www.instagram.com/smart_programming

Java
android
python
Data Science
Machine Learning
Artificial Intelligence
M E A N
Full Stack
BIG DATA
hadoop

InputStream & OutputStream classes in Java

InputStream & OutputStream Diagram :-

=> FileInputStream :-

-> FileInputStream is used to read the data for eg text, image, video, audio etc in the form of bytes

-> FileInputStream can read the character stream data also, but you should use FileReader to read character type of data

-> In this case, if file/data is not found, then it will throw FileNotFoundException

=> FileOutputStream :-

-> FileOutputStream is used to write the data for eg text, image, video, audio etc in the form of bytes

-> FileOutputStream can read the character stream data also but you should use FileWriter to write character stream data

-> In this case if file is not found, then it will create new file

=> ByteArrayInputStream :-

-> ByteArray + InputStream : As name suggests, ByteArrayInputStream can be used to read ByteArray data only as an input stream

-> ByteArrayInputStream uses internal buffer to read the byte array data

=> ByteArrayOutputStream :-

-> ByteArrayOutputStream is used to write the common data to multiple files

=> FilterInputStream :

-> FileInputStream is less used because it provides sub-classes for more functionalities i.e. DataInputStream, BufferedInputStream & PushBackInputStream

=> FilterOutputStream :-

-> FilterOutputStream is less used because it provides sub-class for more functionalities i.e. DataOutputStream, BufferedOutputStream & PrintStream

Smart
Programming
We Educate
We Develop

=> DataInputStream :-

-> DataInputStream is used to read the primitive data from input stream in a machine-independent way

=> DataOutputStream :-

-> DataOutputStream is used to write the primitive data form input stream in a machine-independent way

---

=> BufferedInputStream :-

-> In this case buffer machanish is used internally which will improve the performance

=> BufferedOutputStream :-

-> BufferedOutputStream also use buffer to write the data in order to improve the performance

(flush() method is used forcefully write the data into the stream or output before erasing the buffer memory)

=> PushBackInputStream :

-> PushBackInputStream can be used to unread a byte which is already read and push back one byte

---

=> PrintStream :-

-> PrintStream provides the methods to write the data into another stream

-> PrintStream class automatically flushes the data so there is no need to use flush() method

-> PrintStream method does not throw IOException

=> PipedInputStream & PipedOutputStream :-

-> PipedInputStream and PipedOutputStream are used simultaneously

-> These streams are used to transfer the data from one thread to another thread

-> Both these streams should be connected

=> ObjectInputStream & ObjectOutputStream :-

-> These streams are used in serialization

Company Links & Contacts

Company Name: Smart Programming (+91 62838-30308)

Address : Chandigarh & Mohali (Punjab), India

Websites: https://www.smartprogramming.in/

https://courses.smartprogramming.in

Android App: https://play.google.com/store/apps/details?id=com.smartprogramming

YouTube Channel: https://www.youtube.com/c/SmartProgramming

Smart Programming : YouTube Channel

An investment in Knowledge pays the best interest….

Smart Programming
We Educate - We Develop

☎ +91 62838-30308
Call us to Learn
Latest Technologies

City : Mohali (Punjab),
& Chandigarh
(India)

WEBSITE : http://www.smartprogramming.in

BUY COURSES ON : https://courses.smartprogramming.in

YOUTUBE CHANNEL : Smart Programming (https://www.youtube.com/c/SmartProgramming)

ANDROID APP : Smart Programming
(https://play.google.com/store/apps/details?id=com.smartprogramming)

https://www.facebook.com/smartprogramming.india

https://www.instagram.com/smart_programming

Java
android
python
Data Science
Machine Learning
Artificial Intelligence
M E A N
Full Stack
BIG DATA
hadoop

Reader & Writer classes explanation in Java

=> Reader & Writer Classes Explanation :-

=> InputStreamReader :-

-> InputStreamReader is a bridge which take the input from byte stream and decodes into character stream

=> OutputStreamWriter :-

-> OutputStreamWriter converts the character stream into byte stream using a specified charset

---

=> FileReader :-

-> It inherits the InputStreamReader class

-> FileReader is used to read character stream

=> FileWriter :-

-> FileWriter class inherits OutputStreamWriter class

-> FileWriter is used to write character stream

=> BufferedReader :-

-> BufferedReader is used to read the text from character stream

-> BufferedReader can be used to read data line by line by readLine() method

-> Performance of BufferedReader is fast

=> BufferedWriter :-

-> BufferedWriter is used to write the char array or string into the file

-> BufferedWriter performance is fast

=> LineNumberReader :-

-> LineNumberReader is a buffered character input stream which keeps the track of line numbers

---

=> CharArrayReader :-

-> CharArray + Reader : As the name suggests it is used to read the character array stream

=> CharArrayWriter :-

-> CharArray + Writer : It is used to write the common data to multiple file

---

=> FilterReader :-

-> FilterReader is used to perform filtering operations on reader stream

-> FilterReader is an abstract class

=> FilterWriter :-

-> FilterWriter class is used to write filtered character stream

-> FilterWriter is an abstract class

=> PushbackReader :-

-> PushbackReader is used to push back the one character into stream and read again

---

=> PipedReader :-

-> PipedReader is used to read the content from a pipe as character stream

-> It is used in case of multiple threads

=> PipedWriter :-

-> PipedWriter is used to write the content to a pipe as a character stream

-> It is also used in case of multiple threads

=> StringReader :-

-> StringReader is character stream which string as a source

=> StringWriter :-

-> StringWriter is a character stream which can be used to write data from string as a source

---

Company Links & Contacts

Company Name: Smart Programming (+91 62838-30308)

Address : Chandigarh & Mohali (Punjab), India

Websites: https://www.smartprogramming.in/

https://courses.smartprogramming.in

Android App:
https://play.google.com/store/apps/details?id=com.smartprogramming

YouTube Channel:
https://www.youtube.com/c/SmartProgramming

Smart Programming : YouTube Channel

An investment in Knowledge pays the best interest….

Smart Programming
We Educate - We Develop

☎ +91 62838-30308
Call us to Learn
Latest Technologies

City : Mohali (Punjab),
& Chandigarh
(India)

WEBSITE : http://www.smartprogramming.in

BUY COURSES ON : https://courses.smartprogramming.in

YOUTUBE CHANNEL : Smart Programming (https://www.youtube.com/c/SmartProgramming)

ANDROID APP : Smart Programming
(https://play.google.com/store/apps/details?id=com.smartprogramming)

https://www.facebook.com/smartprogramming.india

https://www.instagram.com/smart_programming

Java
android
python
Data Science
Machine Learning
Artificial Intelligence
M E A N
Full Stack
BIG DATA
hadoop

File Handling in Java

=> File :

-> A file is an object on a computer which stores the data, information, settings, commands etc.

-> There are many file operations for eg. creating new file, rename file, deleting file, writing into file, reading from the file etc

-> There are 2 ways by which we can read data form the file :-

1. Sequential Access : In this way we can read or write the data in sequential form i.e. from beginning or starting of the file
2. Random Access : In this way we can read or write the data in random way i.e. from any location of the file

-> To perform these operations java provided many classes and methods but main class is "java.io.File" class

=> Sequential Access : done in the programs using java.io.File class

=> Random Access :

-> To read and write the files in random way, we have to use "java.io.RandomAccessFile" class

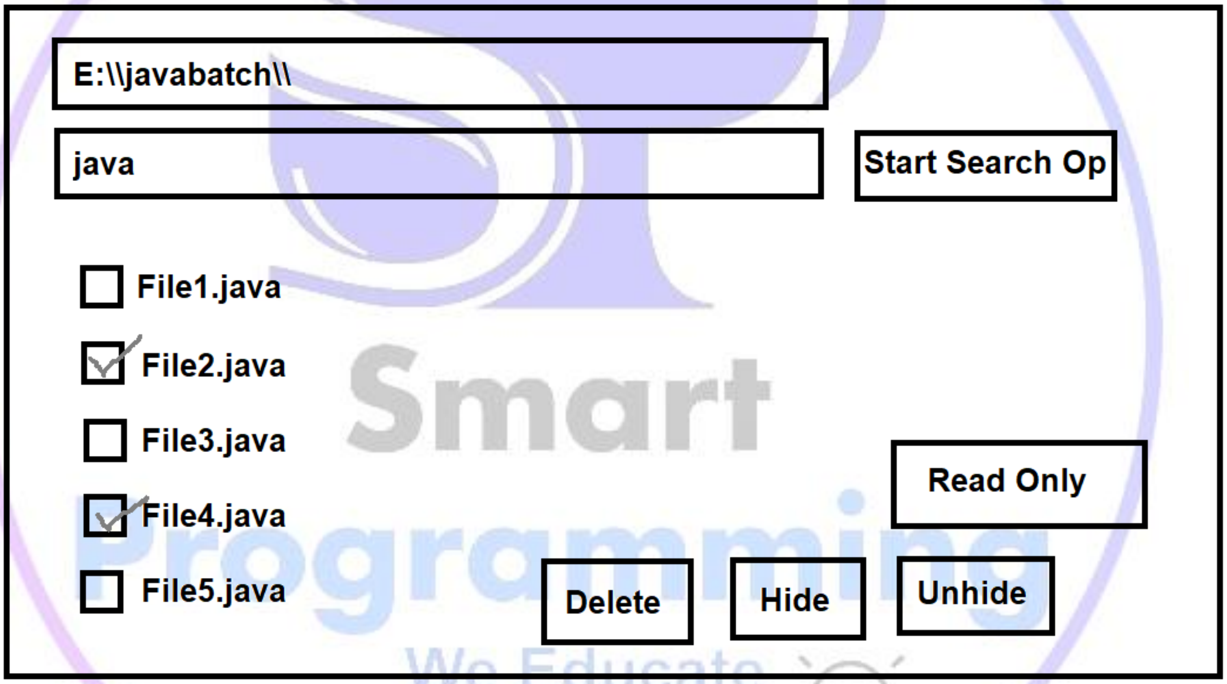
-> There can permissions i.e "r" or "rw"

-> writeXXX(XXX value)

readXXX(XXX value)

seek(long pos)

- Task : Create a frame, search for file and print all the files, then select any file and you will be able to delete, hide, unhide, read only frpm buttons



E:\\javabatch\\

java

☐ File1.java
☑ File2.java
☐ File3.java
☑ File4.java
☐ File5.java

Start Search Op
Read Only
Delete
Hide
Unhide

Company Links & Contacts

Company Name: Smart Programming (+91 62838-30308)

Address : Chandigarh & Mohali (Punjab), India

Websites: https://www.smartprogramming.in/

https://courses.smartprogramming.in

Android App: https://play.google.com/store/apps/details?id=com.smartprogramming

YouTube Channel: https://www.youtube.com/c/SmartProgramming

Smart Programming : YouTube Channel

An investment in Knowledge pays the best interest….

Smart Programming
We Educate - We Develop

☎ +91 62838-30308
Call us to Learn
Latest Technologies

City : Mohali (Punjab),
& Chandigarh
(India)

WEBSITE : http://www.smartprogramming.in

BUY COURSES ON : https://courses.smartprogramming.in

YOUTUBE CHANNEL : Smart Programming (https://www.youtube.com/c/SmartProgramming)

ANDROID APP : Smart Programming
(https://play.google.com/store/apps/details?id=com.smartprogramming)

https://www.facebook.com/smartprogramming.india

https://www.instagram.com/smart_programming

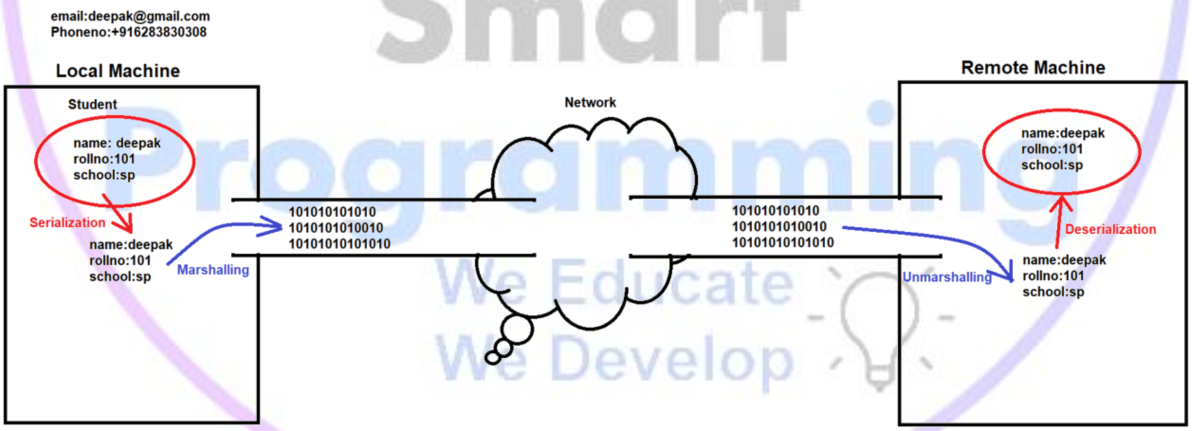
Java
android
python
Data Science
Machine Learning
Artificial Intelligence
M E A N
Full Stack
BIG DATA
hadoop

Serialization & Externalization in Java

=> Serialization & Deserialization :-

-> Serialization is the mechanism by which object state is converted into byte stream and converting byte stream into object state is known as deserialization

-> The byte stream created is platform independent, so the object serialized on one platform can be deserialized on another platform


email:deepak@gmail.com
Phoneno:+916283830308
Local Machine
Student
name: deepak
rollno:101
school:sp
Serialization
name:deepak
rollno:101
school:sp
Marshalling
Network
101010101010
1010101010010
10101010101010
101010101010
1010101010010
10101010101010
Remote Machine
name:deepak
rollno:101
school:sp
Deserialization
Unmarshalling
name:deepak
rollno:101
school:sp

-> Advantages of Serialization and Deserialization :-

1. We can save or persist the state of an object
2. Object can travel in the network

-> How to achieve Serialization & Deserialization

= We have to use following 2 points :-

1. Serializable Interface
2. ObjectOutputStream (Serialization) - writeObject()
   ObjectInputStream (Deserialization) - readObject()

-> Only those objects can be serialized which implements java.io.Serializable interface

=> Serializable Interface :-

-> Serializable interface is a marker interface (marker interface are those which does not have any data member and methods and marker interface does not have any "business logic")

-> It can be used to inject some abilities to the object at runtime

-> More examples of marker interface : Cloneable, Remote etc

-> All the wrapper classes and String class by default inherit Serializable interface

---

=> transient keyword :-

-> If we dont want any variable to serialize then we can create that object as transient

=> Points to remember :-

1. If parent class implements Serializable interface, then we can serialize child class object

2. If we dont want to serialize any variable then we can create that variable as transient or static

3. Constructor of an object is not called when an object is deserialized

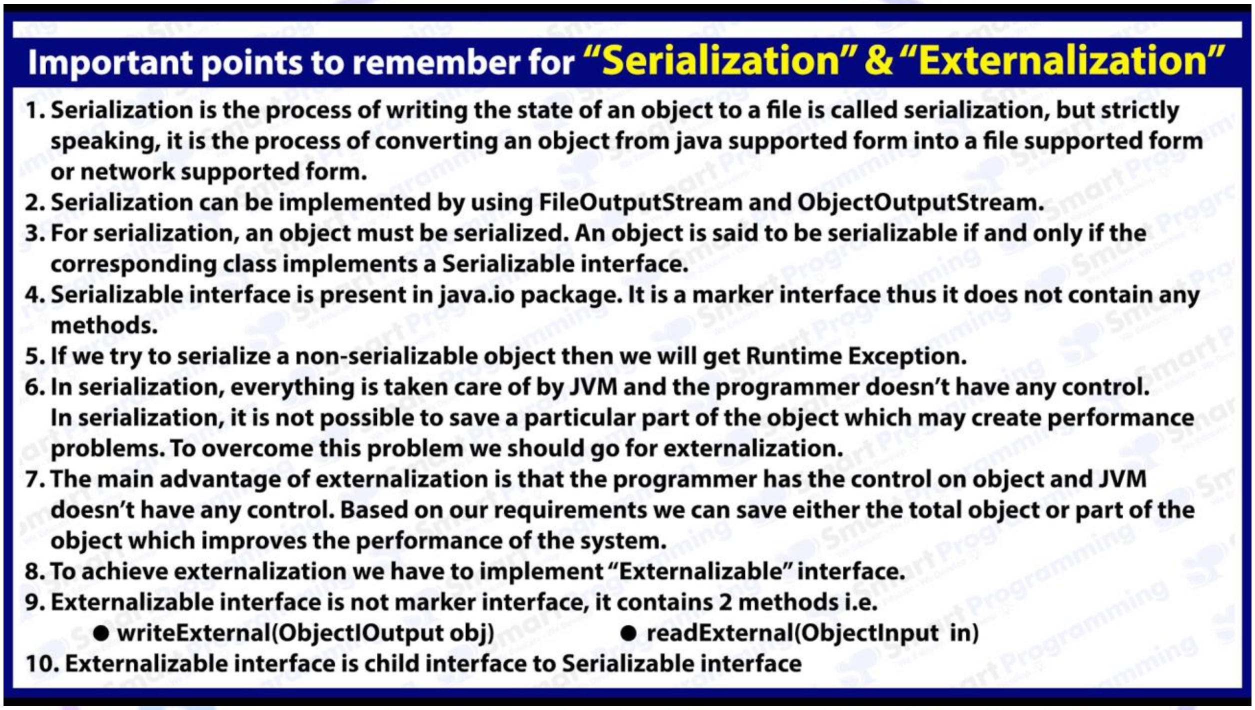
---

=> Externalization :-

-> Externalization is when we need customized serialization

-> How to achieve Externalization :-

1. java.io.Externalizable interface
    - -> writeExternal(-)
    - -> readExternal(-)



Important points to remember for "Serialization" & "Externalization"

1. Serialization is the process of writing the state of an object to a file is called serialization, but strictly speaking, it is the process of converting an object from java supported form into a file supported form or network supported form.
2. Serialization can be implemented by using FileOutputStream and ObjectOutputStream.
3. For serialization, an object must be serialized. An object is said to be serializable if and only if the corresponding class implements a Serializable interface.
4. Serializable interface is present in java.io package. It is a marker interface thus it does not contain any methods.
5. If we try to serialize a non-serializable object then we will get Runtime Exception.
6. In serialization, everything is taken care of by JVM and the programmer doesn't have any control. In serialization, it is not possible to save a particular part of the object which may create performance problems. To overcome this problem we should go for externalization.
7. The main advantage of externalization is that the programmer has the control on object and JVM doesn't have any control. Based on our requirements we can save either the total object or part of the object which improves the performance of the system.
8. To achieve externalization we have to implement "Externalizable" interface.
9. Externalizable interface is not marker interface, it contains 2 methods i.e.
    - writeExternal(ObjectIOutput obj)
    - readExternal(ObjectInput in)
10. Externalizable interface is child interface to Serializable interface

Company Links & Contacts

Company Name: Smart Programming (+91 62838-30308)

Address : Chandigarh & Mohali (Punjab), India

Websites: https://www.smartprogramming.in/

https://courses.smartprogramming.in

Android App:
https://play.google.com/store/apps/details?id=com.smartprogramming

YouTube Channel:
https://www.youtube.com/c/SmartProgramming

Smart Programming : YouTube Channel

An investment in Knowledge pays the best interest….

Smart Programming
We Educate - We Develop

☎ +91 62838-30308
Call us to Learn
Latest Technologies

City : Mohali (Punjab),
& Chandigarh
(India)

WEBSITE : http://www.smartprogramming.in

BUY COURSES ON : https://courses.smartprogramming.in

YOUTUBE CHANNEL : Smart Programming (https://www.youtube.com/c/SmartProgramming)

ANDROID APP : Smart Programming
(https://play.google.com/store/apps/details?id=com.smartprogramming)

https://www.facebook.com/smartprogramming.india

https://www.instagram.com/smart_programming

Java
android
python
Data Science
Machine Learning
Artificial Intelligence
M E A N
Full Stack
BIG DATA
hadoop

Networking Introduction

=> Networking :-

➔ Networking is the process of connecting two or more systems in order to achieve data sharing or resource sharing.

➔ Types of networking :-

1. LAN (Local Area Network) - Small building like offices, college labs etc
2. PAN (Personal Area Network) - For personal use within 10-12 meters
3. MAN (Metropolitian Area Network) - It is within the city like banks within the city
4. WAN (Wide Area Network) - It is within the states or countries

➔ To establish networking we needs basically 3 things :-

1. Hardware :- systems, routers, switch, hub etc
2. Media :- fibre optics, wires, wireless etc
3. Software :- any programming language (java)

=> Terms used in networking :-

1. IP Address :-

- ➔ IP Address full form is Internet Protocol Address
- ➔ IP address is the unique integer or decimal value which is provided by ISP (Internet Service Provider) whenever the system is connected to the network
- ➔ IP Address allows the computer to send or receive data
- ➔ IP Address confirms to send or receive the data to the accurate system
- ➔ IP Address are of 4 types :-
  public, private, static, dynamic
- ➔ Command to get the IP Address of my system :-
  ipconfig or ipconfig/all

2. Port Number :-

- ➔ Port number is the unique integer value for each and every service or process which is executing in the system
- ➔ For example : 80 - HTTP, 3306 - MySQL etc
- ➔ Command to get the Port number :- netstat -ano

3. Protocol :-

- ➔ It is the set of rules and regulations for data transfer or communication
- ➔ There are a lot of protocols for example :- IP (Internet Protocol), TCP (Transmission Control Protocol), UDP (User Datagram Protocol), HTTP (Hyper Text Transfer Protocol), FTP (File Transfer Protocol), SMTP (Simple Mail Transfer Protocol), POP3 (Post Office Protocol 3), SSH (Secure Shell) etc

➔ NOTE :-

1. Java provides a package i.e. java.net for TCP and UDP
2. Classes used for TCP : Socket, ServerSocket etc
3. Classes used for UDP : DatagramPacket, DatagramSocket etc
4. Other important classes used for networking are : URL, URLCOnnection, HTTPConnection etc

4. MAC Address :-

➔ MAC Address is the unique integer value for the machine which is embedded in the Network Card (Network Interface Card) at the time of manufacturing

➔ MAC Address cannot be changed (But we can change the IP Address)

➔ MAC Address is also known as Physical Address

5. Socket :-

- ➔ Socket is the end-point of the communication

- ➔ Socket is the buffer which is the combination of IP Address, Port Number & Protocol
- ➔ Socket is an object from which we can read or write the data

---

=> What is difference between IP Address and Port Number ?

-> IP address is the unique integer value of the system in the network

Port number is the unique integer value of the services or process running is the current/present system

=> What is difference between TCP & UDP ?

1. TCP :- It is connection oriented

   UDP :- It is connection-less

2. TCP :- It is fast

   UDP :- It is slow

=> What is difference between IP Address & MAC Address ?

- ➔ IP address is the unique integer value of the system in the network
  MAC Address is the unique integer value for the system which is embedded in the Network Card (Network Interface Card) at the time of manufacturing

- ➔ IP Address is provided by ISP
MAC Address is provided by the manufacturing companies

- ➔ IP Address can be changed
MAC Address cannot be changed

=> We can create 2 types of applications in java :-

1. Standalone Applications (Desktop Applications) :- Standalone applications are those which does not use client-server architecture and they are executed only on one system. For examples :- TicTacToe game, Editor (Notepad), Billing Software, Car-Racing Game etc

2. Distributed Applications (Client-Server Project) :- Distributed applications are those which use client-server architecture and they can communicate with multiple systems. For examples :- Chat Server, Banking etc

=> Note :-

➔ To prepare distributed applications java has provided multiple technologies i.e.

1. Socket Programming
2. RMI
3. CORBA
4. EJB
5. Web Services

etc

Company Links & Contacts

Company Name: Smart Programming (+91 62838-30308)

Address : Chandigarh & Mohali (Punjab), India

Websites: https://www.smartprogramming.in/
https://courses.smartprogramming.in

Android App:
https://play.google.com/store/apps/details?id=com.smartprogramming

YouTube Channel:
https://www.youtube.com/c/SmartProgramming

Smart Programming : YouTube Channel

An investment in Knowledge pays the best interest….

Smart Programming
We Educate - We Develop

☎ +91 62838-30308
Call us to Learn
Latest Technologies

City : Mohali (Punjab), & Chandigarh (India)

WEBSITE : http://www.smartprogramming.in

BUY COURSES ON : https://courses.smartprogramming.in

YOUTUBE CHANNEL : Smart Programming (https://www.youtube.com/c/SmartProgramming)

ANDROID APP : Smart Programming
(https://play.google.com/store/apps/details?id=com.smartprogramming)

https://www.facebook.com/smartprogramming.india

https://www.instagram.com/smart_programming

Java
android
python
Data Science
Machine Learning
Artificial Intelligence
M E A N
Full Stack
BIG DATA
hadoop

Socket Programming

=> Socket Programming :-

- ➔ Socket Programming is used for communication between two systems (two JRE)
- ➔ Socket Programming can be connection-oriented or connection-less
- ➔ To achieve Socket Programming java has provided one package java.net package

- ➔ For connection-oriented TCP is used and java provides some classes i.e. ServerSocket & Socket
- ➔ For connection-less UDP is used and java provides some classes i.e. DatagramPacket & DatagramSocket
- ➔ These all classes are present in java.net package

- ➔ Socket Programming is used to create distributed applications

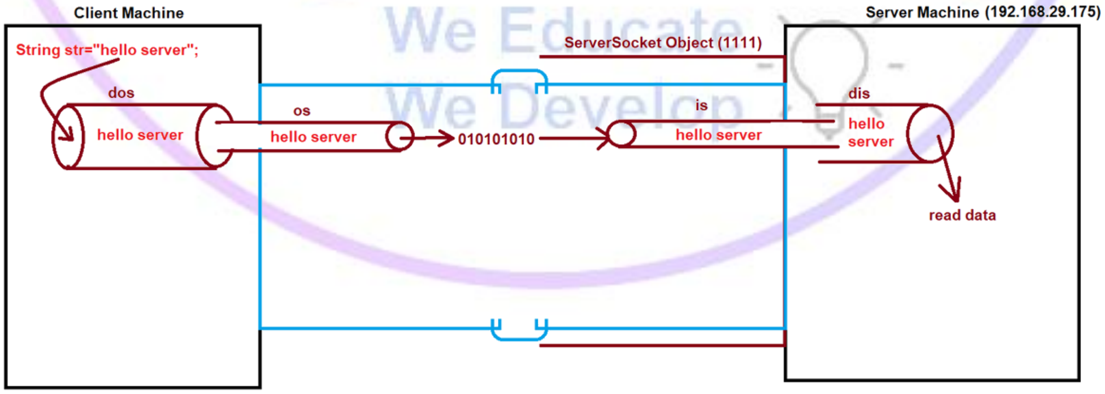
=> Steps for Socket Programming :-

➔ Steps for Server side -

1. Create ServerSocket object and run it on certain port number
2. Call accept() method of ServerSocket class. It will wait for client request and as soon as client sends the request, it will accept the request

➔ Steps for Client side -

1. Create Socket object on client side and provide IP Address and Port Number


Client Machine
Server Machine (192.168.29.175)
String str="hello server";
dos
hello server
os
hello server
010101010
ServerSocket Object (1111)
is
hello server
dis
hello
server
read data

Company Links & Contacts

Company Name: Smart Programming (+91 62838-30308)

Address : Chandigarh & Mohali (Punjab), India

Websites: https://www.smartprogramming.in/

https://courses.smartprogramming.in

Android App: https://play.google.com/store/apps/details?id=com.smartprogramming

YouTube Channel: https://www.youtube.com/c/SmartProgramming

Smart Programming : YouTube Channel

An investment in Knowledge pays the best interest….

Smart Programming
We Educate - We Develop

☎ +91 62838-30308
Call us to Learn
Latest Technologies

City : Mohali (Punjab),
& Chandigarh
(India)

WEBSITE : http://www.smartprogramming.in

BUY COURSES ON : https://courses.smartprogramming.in

YOUTUBE CHANNEL : Smart Programming (https://www.youtube.com/c/SmartProgramming)

ANDROID APP : Smart Programming
(https://play.google.com/store/apps/details?id=com.smartprogramming)

https://www.facebook.com/smartprogramming.india

https://www.instagram.com/smart_programming

Java
android
python
Data Science
Machine Learning
Artificial Intelligence
M E A N
Full Stack
BIG DATA
hadoop

Remote Method Invocation

=> RMI :- (Remote Method Invocation)

- ➔ RMI is an API which is used to create Distributed Applications in java
- ➔ RMI API allows an object to invoke another method that exists in another address space or another JVM
- ➔ RMI provides remote communication between the applications using two objects i.e. "stub" & "skeleton".

  - ⇨ NOTE : RMI API is provided in "java.rmi" package

=> Stub object :-

- ➔ Stub is an object which acts as a gateway for the client side. All the outgoings request are routed through stub object.

➔ Stub object has the following tasks :-

1. It initiates the connection with remote machine (remote JVM)
2. It writes and transmits the parameters from client to server (remote machine)
3. It will wait for the result
4. It can read the returned result/expression or exception
5. Finally, it can return the value to the caller/client

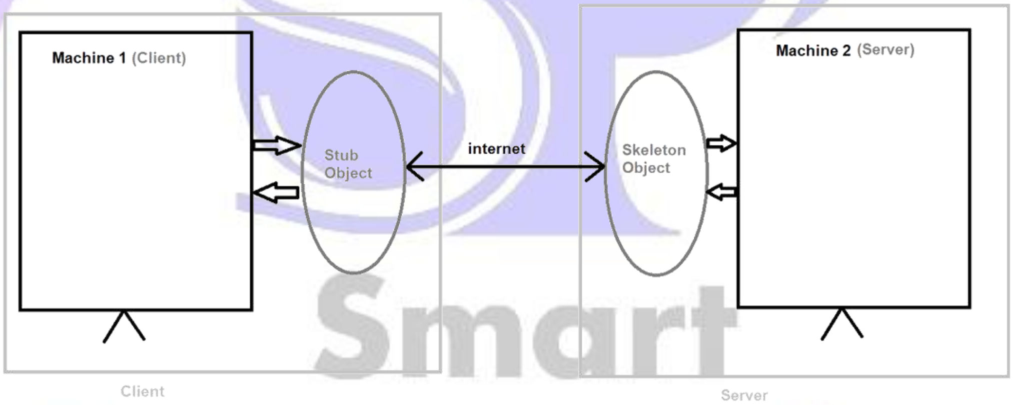
=> Skeleton Object :-

➔ Skeleton is an object which acts as a gateway for the server side. All the incomming requests are routed through skeleton object

➔ Skeleton object has the following tasks :-

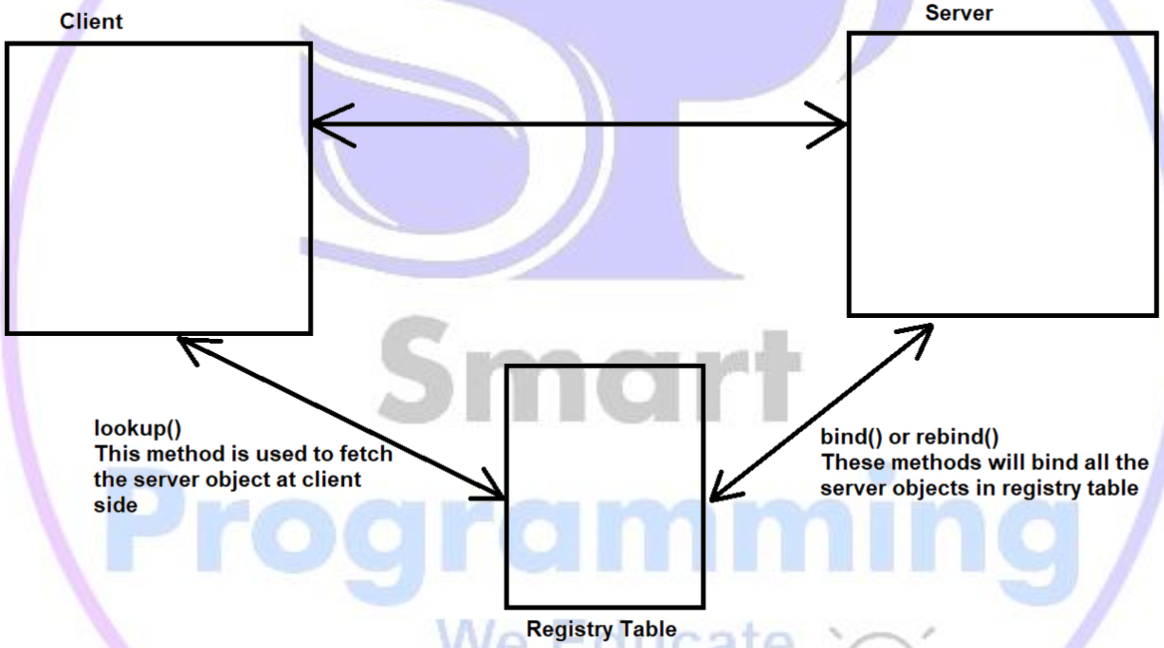
1. It can read the parameters for remote machine
2. It will invoke the remote method
3. It can write and transmits the result

⇨ NOTE :- From rmic 1.2 version, skeleton object is not generated. New JRMP (Java Remote Method Protocol) has got rid of skeleton object


Machine 1 (Client)
Stub Object
internet
Skeleton Object
Machine 2 (Server)
Client
Server

=> RMI Registry :-

-> It is the namespace on which all the server objects are placed


Client
Server
lookup()
This method is used to fetch the server object at client side
bind() or rebind()
These methods will bind all the server objects in registry table
Registry Table

=> Steps for RMI program :-

1. Create the remote interface
2. Create an implemented class for remote interface
3. Create client and server
4. Compile, stub & skeleton (rmic)
5. start the registry (rmiregistry)
6. Start the server
7. Run the client

Company Links & Contacts

Company Name: Smart Programming (+91 62838-30308)

Address : Chandigarh & Mohali (Punjab), India

Websites: https://www.smartprogramming.in/

https://courses.smartprogramming.in

Android App:
https://play.google.com/store/apps/details?id=com.smartprogramming

YouTube Channel:
https://www.youtube.com/c/SmartProgramming

Smart Programming : YouTube Channel

An investment in Knowledge pays the best interest....

Smart Programming

We Educate - We Develop

+91 62838-30308

Call us to Learn Latest Technologies

City : Mohali (Punjab), & Chandigarh (India)

WEBSITE : http://www.smartprogramming.in

BUY COURSES ON : https://courses.smartprogramming.in

YOUTUBE CHANNEL : Smart Programming (https://www.youtube.com/c/SmartProgramming)

ANDROID APP : Smart Programming (https://play.google.com/store/apps/details?id=com.smartprogramming)

https://www.facebook.com/smartprogramming.india

https://www.instagram.com/smart_programming

Java
android
python
Data Science
Machine Learning
Artificial Intelligence
M E A N
Full Stack
BIG DATA
hadoop

Data Structure & Arrays in Java

=> Data Structure :-

-> A data structure is a particular way by which we organize, manage and store the data in the computer so that it can be used effectively

-> Data structure is not a programming langaue like c, c++, java or python. It is a set of algorithms which we can implment in any programming langauges

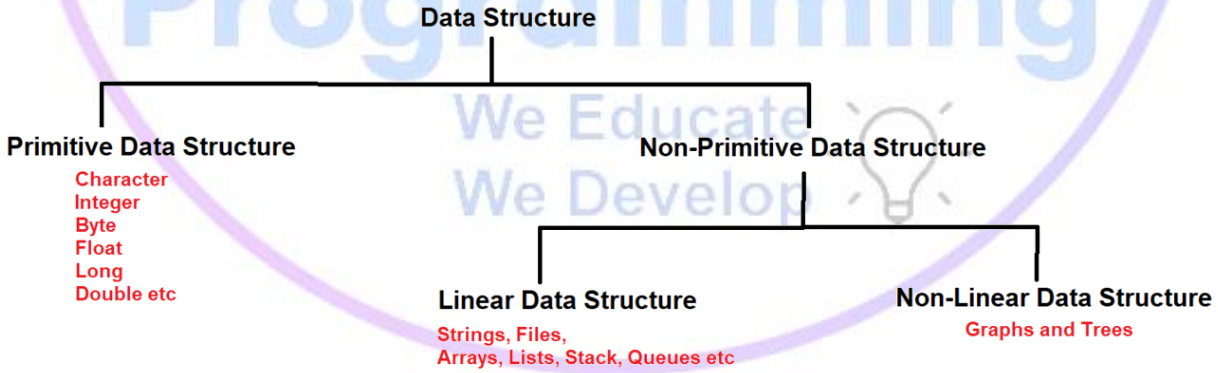
-> Operations that we can perform on data :-

1. Searching
2. Sorting
3. Insertion
4. Deletion
5. Updation

-> Advantages :-

1. We can process the data easily
2. It stores the data very efficiently on the disk
3. Data structure is used to develop algorithms
4. It stores the data in a secured way

-> Types of Data Structure :-


Data Structure
Primitive Data Structure
Character
Integer
Byte
Float
Long
Double etc
Non-Primitive Data Structure
Linear Data Structure
Strings, Files,
Arrays, Lists, Stack, Queues etc
Non-Linear Data Structure
Graphs and Trees

1. Primitive Data Structure (eg Integer, Float, Long, Double etc)

2. Non-Primitive Data Structure

-> Linear DS (eg String, Files, Arrays, Lists, Stacks, Queues etc)

= In Linear DS the data is arranged in a sequential form i.e. one element is connected to only one other element in linear form

-> Non-Linear DS (eg Graphs & Trees)

= In non-linear DS, one element can connect with 'n' number of elements.

=> Arrays :-

1. Array is a collection of similar type of data (homogeneous data)

2. Array elements are stored in contigeous memory locations

3. Array can contain primitive or non-primitive elements

4. Array is index based data structure, first index position of an array is 0

5. Array length starts from 1

Company Links & Contacts

Company Name: Smart Programming (+91 62838-30308)

Address : Chandigarh & Mohali (Punjab), India

Websites: https://www.smartprogramming.in/

https://courses.smartprogramming.in

Android App:
https://play.google.com/store/apps/details?id=com.smartprogramming

YouTube Channel:
https://www.youtube.com/c/SmartProgramming

Smart Programming : YouTube Channel

An investment in Knowledge pays the best interest….

Smart Programming
We Educate - We Develop

☎ +91 62838-30308
Call us to Learn Latest Technologies

City : Mohali (Punjab), & Chandigarh (India)

WEBSITE : http://www.smartprogramming.in

BUY COURSES ON : https://courses.smartprogramming.in

YOUTUBE CHANNEL : Smart Programming (https://www.youtube.com/c/SmartProgramming)

ANDROID APP : Smart Programming (https://play.google.com/store/apps/details?id=com.smartprogramming)

https://www.facebook.com/smartprogramming.india

https://www.instagram.com/smart_programming

Java
android
python
Data Science
Machine Learning
Artificial Intelligence
M E A N
Full Stack
BIG DATA
hadoop

Single Dimensional Array in Java

=> Types of Arrays :-

1. Single Dimensional Array
    - -> 1D Array
2. Multi-Dimensional Array
    - -> 2D Array
    - -> 3D Array
    - -> 4D, 5D, 6D.... Array
    - -> Zic-Zac Array
3. Anonymous Array

=> Single Dimensional Array (for example 1D Array) :-

-> In this type of array, there is only one row or one column

-> Following points for each type of array :-

= Declaration :

1. Array can be declared normally like simple variables but we have to provide square([]) braces

2. When we declare an array, we dont need to provide the size of an array

= Creation :-

1. When we create an array by using new keyword, we have to provide the size of an array

2. When we create an array by using new keyword, all the index position will be initialized by its default values

=> We can declare and create an array within a single line

=Initialization :-

1. We can provide the values at particular index position

=> We can declare, create and initlize an array within single line

= Retrieve :-

1. We can retrieve the value by using for loop

⇨ Different cases for Declaration, Creation & Initialization of an array

=> Declaration :-

1. int[] a,b; //a and b are both arrays
2. int []a,b; //a and b both are arrays
3. int a[],b; //correct, a is an array but b is simple variable
4. int a,b[]; //correct, a is normal variable but b is an array
5. int a[], b[]; //both a and b are array
6. int []a, b[]; //a and b are both arrays

=> Creation :-

1. a=new int[5]; //correct
2. a=new int[]; //error

3. int[] a=new int[5]; //correct

4. int a[]=new int[5]; //correct

5. int []a=new [5]int; //error

6. int[] a=new int[0]; //it will successfully compile and run

7. int[] a=new int[-3]; //it will compile but provides runtime exception i.e. java.lang.NegativeArraySizeException

=> Initialization :-

1. int[] a=new int[3];

a[3]=100; //compile successfully but will throw runtime exception saying ArrayIndexOutOfBoundsException

2. If we dont initialize any proper index position value, then it will compile and run successfully

3. a[-1]=100; //compile successfully but will provide runtime exception saying ArrayIndexOutOfBoundsException

---

-> Points to remember :-

1. Array are Objects in java
2. Arrays are stored in "Heap Area"

Interview Questions :-

1. Difference between declaring, creating and initializing an array with and without using new keyword

-> When we create an array without using new keyword then default value will not be initialized in the array, all the values will be initlized automatically in the array

Company Links & Contacts

Company Name: Smart Programming (+91 62838-30308)

Address : Chandigarh & Mohali (Punjab), India

Websites: https://www.smartprogramming.in/

https://courses.smartprogramming.in

Android App: https://play.google.com/store/apps/details?id=com.smartprogramming

YouTube Channel: https://www.youtube.com/c/SmartProgramming

Smart Programming : YouTube Channel

An investment in Knowledge pays the best interest....

Smart Programming
We Educate - We Develop

+91 62838-30308
Call us to Learn
Latest Technologies

City : Mohali (Punjab),
& Chandigarh
(India)

WEBSITE : http://www.smartprogramming.in

BUY COURSES ON : https://courses.smartprogramming.in

YOUTUBE CHANNEL : Smart Programming (https://www.youtube.com/c/SmartProgramming)

ANDROID APP : Smart Programming
(https://play.google.com/store/apps/details?id=com.smartprogramming)

https://www.facebook.com/smartprogramming.india

https://www.instagram.com/smart_programming

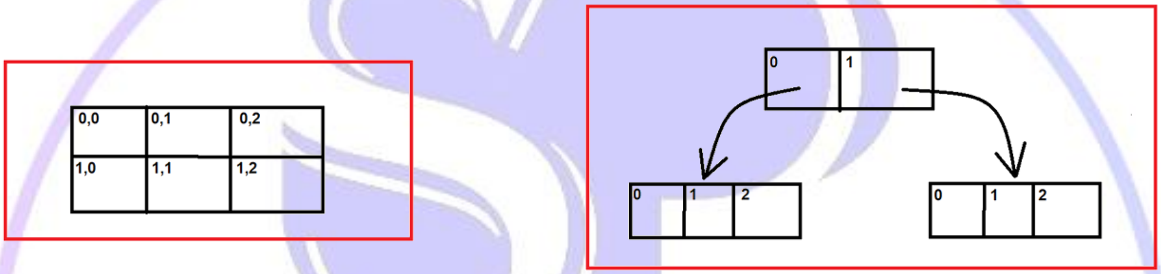
Java
android
python
Data Science
Machine Learning
Artificial Intelligence
M E A N
Full Stack
BIG DATA
hadoop

Two Dimensional Array in Java

=> Types of Arrays :-

1. Single Dimensional Array
   - -> 1D Array
2. Multi-Dimensional Array
   - -> 2D Array
   - -> 3D Array
   - -> 4D, 5D, 6D.... Array
   - -> Zic-Zac Array
3. Anonymous Array

=> 2D Array :-



2 Dimensional Array Represenation

-> Following points for 2D array :-

= Declaration

-> For declaring 2D array, we have to use double square braces

= Creation :-

-> When we create 2D array, we have to provide the size of an array

-> a=new int[2][3]; // there are 2 rows and 3 columns

-> Whenever we create an array by using new keyword, then default values will be initialized in the array blocks

-> We can declare and create 2D array in a single line

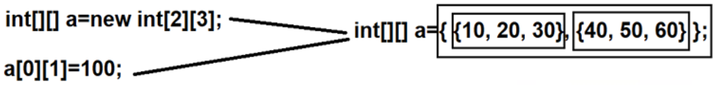
int[][] a=new int[2][3];

= Initialization

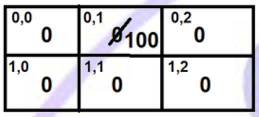
-> a[0][1]=100; //will initialize 100 value at 0,1 index position

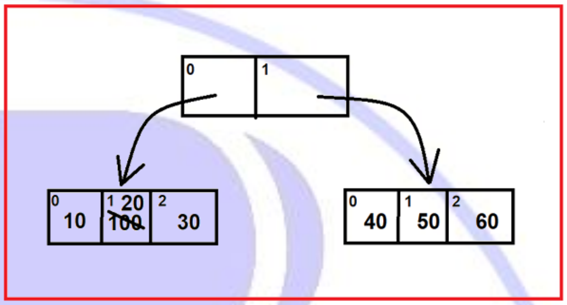
-> We can declare, create and initialize an array in single line

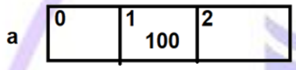
int[][] a={{10,20,30}, {40,50,60}};



```
int[][] a=new int[2][3];
a[0][1]=100;
int[][] a={ {10, 20, 30}, {40, 50, 60} };
```


0,0
0
0,1
~~0~~ 100
0,2
0
1,0
0
1,1
0
1,2
0


0
1
0
10
1
20
~~100~~
2
30
0
40
1
50
2
60

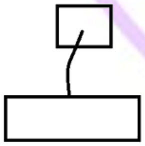


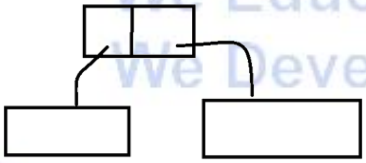
```
a[1]=100;
int[] a={10,20,30};
```

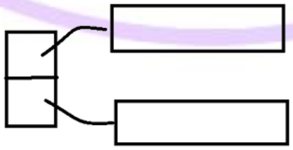
`int[][] a={ { } };`

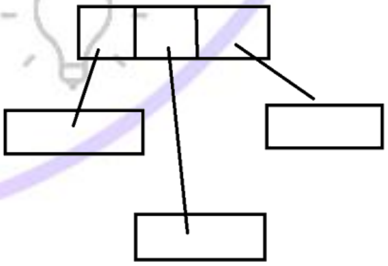
`int[][] a={ { }, { } };`

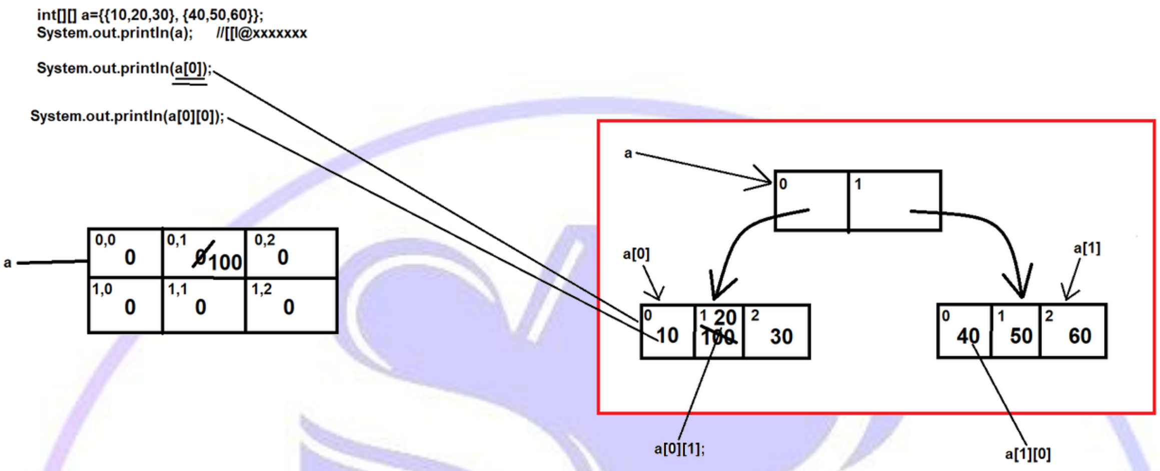
`int[][] a={ { }, { }, { } };`











```
int[][] a={{10,20,30}, {40,50,60}};
System.out.println(a);     //[[I@xxxxxxx

System.out.println(a[0]);

System.out.println(a[0][0]);
```

a
0,0
0
0,1
0
100
0,2
0
1,0
0
1,1
0
1,2
0
a
0
1
a[0]
a[1]
0
10
1
20
100
2
30
0
40
1
50
2
60
a[0][1];
a[1][0]

= Retrieve

-> We can retrieve elements by using for loop

=> Declaration :-

1. int[][] a; //prefered way
2. int [][]a;
3. int[][]a;
4. int [][] a;
5. int []a[];

=> Cases for Declaration :-

1. int [][]a, b; //a (2D array); b (2D array)
2. int a[][], b; //a (2D array); b (simple variable)
3. int []a[], b; //a (2D array); b (1D array)
4. int []a[], b[]; //a (2D array); b (2D array)

5. int [][]a, b[]; //a (2D array); b (3D array)

6. int [][]a, []b; //compile time error

=> Cases for Creation :-

1. a=new int[2][3]; //correct

2. a=new int[][]; //compile time error (array dimension missing)

3. a=new int[2][]; //correct

4. a=new int[][3]; //compile time error (']' expected)

=> Cases for creation within single line :-

1. int[][] a=new int[2][3]; //correct
2. int[][] a=new int[2][]; //correct
3. int[][] a=new int[][]; //error
4. int[][] a=new int[0][0]; //correct
5. int[][] a=new int[-2][3];//will compile successfully but provides runtime exception saying java.lang.NegativeArraySizeException

Company Links & Contacts

Company Name: Smart Programming (+91 62838-30308)

Address : Chandigarh & Mohali (Punjab), India

Websites: https://www.smartprogramming.in/

https://courses.smartprogramming.in

Android App:
https://play.google.com/store/apps/details?id=com.smartprogramming

YouTube Channel:
https://www.youtube.com/c/SmartProgramming

Smart Programming : YouTube Channel

An investment in Knowledge pays the best interest....

Smart Programming
We Educate - We Develop

+91 62838-30308
Call us to Learn
Latest Technologies

City : Mohali (Punjab), & Chandigarh (India)

WEBSITE : http://www.smartprogramming.in

BUY COURSES ON : https://courses.smartprogramming.in

YOUTUBE CHANNEL : Smart Programming (https://www.youtube.com/c/SmartProgramming)

ANDROID APP : Smart Programming (https://play.google.com/store/apps/details?id=com.smartprogramming)

https://www.facebook.com/smartprogramming.india

https://www.instagram.com/smart_programming

Java
android
python
Data Science
Machine Learning
Artificial Intelligence
M E A N
Full Stack
BIG DATA
hadoop

Three Dimensional Array in Java

=> Types of Arrays :-

1. Single Dimensional Array
    - -> 1D Array
2. Multi-Dimensional Array
    - -> 2D Array
    - -> 3D Array
    - -> 4D, 5D, 6D.... Array
    - -> Zic-Zac Array
3. Anonymous Array

=> 3D Array :-

-> Diagram representation

-> How to declare, create and initialize 3D array

= Declaration :-

-> We can declare 3D array by using 3 square braces.

= Creation :-

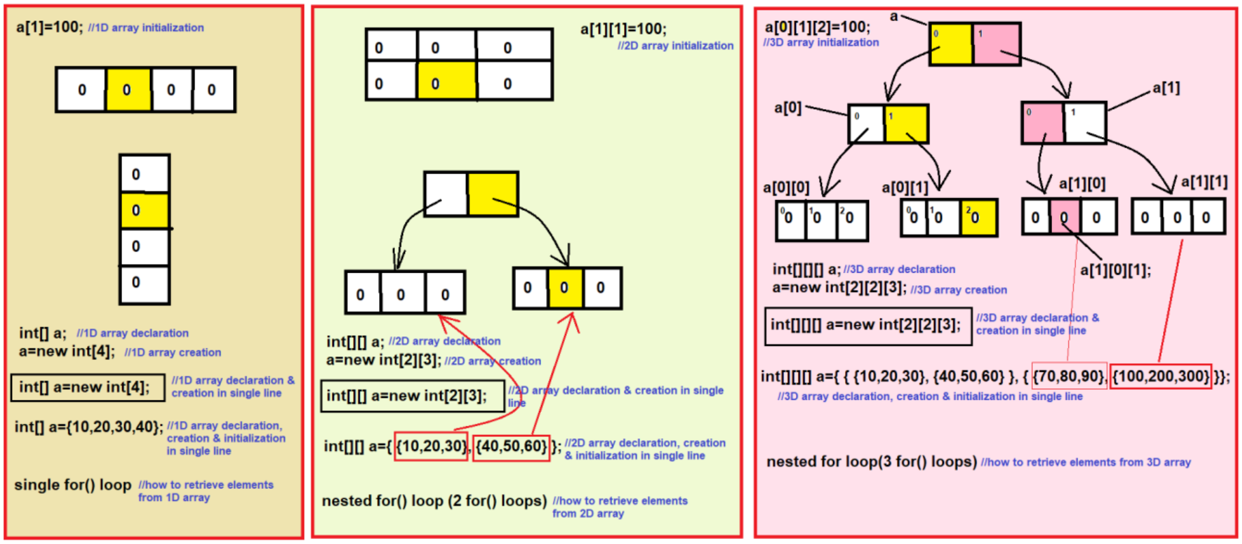
-> We can create an array by using new keyword

= Initialization :-

a[0][1][2]=100;

= Retrieve :-

We can retrieve the elements of 3D array by using 3 for loops



```
a[1]=100; //1D array initialization

int[] a; //1D array declaration
a=new int[4]; //1D array creation

int[] a=new int[4]; //1D array declaration & creation in single line

int[] a={10,20,30,40}; //1D array declaration, creation & initialization in single line

single for() loop //how to retrieve elements from 1D array
```

```
a[1][1]=100; //2D array initialization

int[][] a; //2D array declaration
a=new int[2][3]; //2D array creation

int[][] a=new int[2][3]; //2D array declaration & creation in single line

int[][] a={ {10,20,30}, {40,50,60} }; //2D array declaration, creation & initialization in single line

nested for() loop (2 for() loops) //how to retrieve elements from 2D array
```

```
a[0][1][2]=100; //3D array initialization

int[][][] a; //3D array declaration
a=new int[2][2][3]; //3D array creation

int[][][] a=new int[2][2][3]; //3D array declaration & creation in single line

int[][][] a={ { {10,20,30}, {40,50,60} }, { {70,80,90}, {100,200,300} }};
//3D array declaration, creation & initialization in single line

nested for loop(3 for() loops) //how to retrieve elements from 3D array
```

Declaration :-

1. int[][][] a;
2. int[] [][]a;
3. int [][][]a;
4. int []a[][];
5. int [][]a[];
6. int a[][][];

Cases for Declaration :-

1. int[][][]a, b; //a & b are 3D array
2. int [][]a[],b; //a is 3D array & b is 2D array
3. int [][]a[], b[]; //a and b are 3D array

4. int [][][]a, b[]; //a is 3D array & b is 4D array
5. int [][][]a, []b; //error

Cases for creation :-

1. a=new int[2][2][3]; //correct
2. a=new int[2][2][]; //correct
3. a=new int[2][][]; //correct
4. a=new int[][][]; //error
5. a=new int[2][][3]; //error

Company Links & Contacts

Company Name: Smart Programming (+91 62838-30308)

Address : Chandigarh & Mohali (Punjab), India

Websites: https://www.smartprogramming.in/

https://courses.smartprogramming.in

Android App: https://play.google.com/store/apps/details?id=com.smartprogramming

YouTube Channel: https://www.youtube.com/c/SmartProgramming

Smart Programming : YouTube Channel

An investment in Knowledge pays the best interest….

Smart Programming
We Educate - We Develop

+91 62838-30308
Call us to Learn
Latest Technologies

City : Mohali (Punjab), & Chandigarh (India)

WEBSITE : http://www.smartprogramming.in

BUY COURSES ON : https://courses.smartprogramming.in

YOUTUBE CHANNEL : Smart Programming (https://www.youtube.com/c/SmartProgramming)

ANDROID APP : Smart Programming (https://play.google.com/store/apps/details?id=com.smartprogramming)

https://www.facebook.com/smartprogramming.india

https://www.instagram.com/smart_programming

Java
android
python
Data Science
Machine Learning
Artificial Intelligence
M E A N
Full Stack
BIG DATA
hadoop

Matrix, Jagged & Anonymous Arrays; Arrays Class in Java

=> Types of Arrays :-

1. Single Dimensional Array
    - -> 1D Array
2. Multi-Dimensional Array
    - -> 2D Array
    - -> 3D Array
    - -> 4D, 5D, 6D.... Array
    - -> Zic-Zac Array
3. Anonymous Array

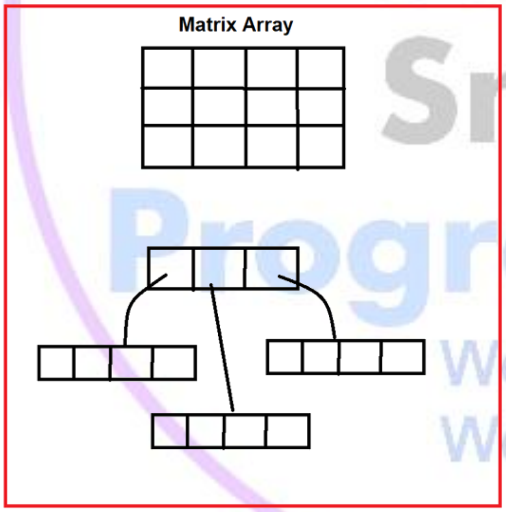
Smart
Programming
We Educate
We Develop

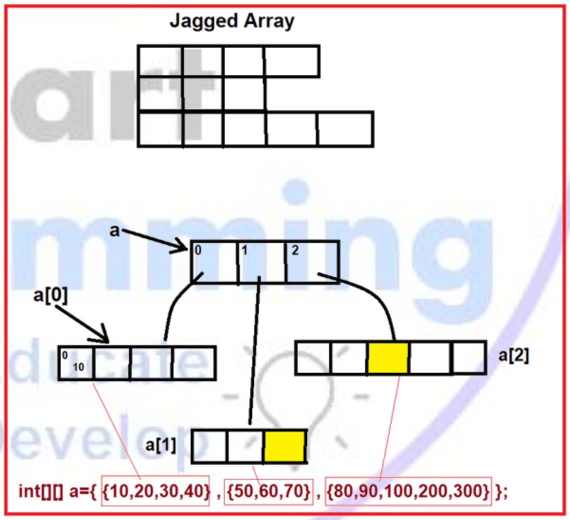
=> Multi-Dimensional Array :

-> Multi-Dimensional array can be Matrix Array or Jagged Array

-> Matrix Array : The array in which there is fixed number of rows & columns

-> Jagged Array : The array in which number of columns can be changed


Matrix Array


Jagged Array
a
0
1
2
a[0]
0
10
a[1]
a[2]

```
int[][] a={ {10,20,30,40} , {50,60,70} , {80,90,100,200,300} };
```

=> Anonymous Array :-

-> An array that does not have any name is known as Anonymous Array

-> For example :

`new int[]{10,20,30,40};`

-> Use : Anonymous arrays are created when we want to use them instantly or only for single time

-> Anonymous arrays are used as an argument in the method

-> Anonymous array can be single dimensional or multi-dimensional array

=> Different ways to print/iterate an array elements :-

-> for loop

-> while loop

-> for-each loop

```
for(type var:array)
{
    //print(var)
}
```

-> Limitations of for-each loop :-

1. It does not track the index position of an array
2. It only iterates the elements in forward direction
3. It cannot be used to modify the array

=> Arrays class :-

-> Java has created an "Arrays" class in "java.util" package

-> Arrays class directly inherits the Object class

-> "Arrays" class provides the static methods only

Company Links & Contacts

Company Name: Smart Programming (+91 62838-30308)

Address : Chandigarh & Mohali (Punjab), India

Websites: https://www.smartprogramming.in/

https://courses.smartprogramming.in

Android App: https://play.google.com/store/apps/details?id=com.smartprogramming

YouTube Channel: https://www.youtube.com/c/SmartProgramming

Smart Programming : YouTube Channel

An investment in Knowledge pays the best interest….

Smart Programming
We Educate - We Develop

+91 62838-30308
Call us to Learn
Latest Technologies

City : Mohali (Punjab),
& Chandigarh
(India)

WEBSITE : http://www.smartprogramming.in

BUY COURSES ON : https://courses.smartprogramming.in

YOUTUBE CHANNEL : Smart Programming (https://www.youtube.com/c/SmartProgramming)

ANDROID APP : Smart Programming
(https://play.google.com/store/apps/details?id=com.smartprogramming)

https://www.facebook.com/smartprogramming.india

https://www.instagram.com/smart_programming

Java
android
python
Data Science
Machine Learning
Artificial Intelligence
M E A N
Full Stack
BIG DATA
hadoop

Collection Framework in Java

=> Types Of Data Structure :-

1. Primitive Data Structure

Examples are : boolean, char, byte, short, int, long, float, double

2. Non-Primitive Data Structure

Examples are : String, Files, Arrays, Collection Framework (Collection, Map)

=> Applications of Arrays :-

1. Arrays are used to store multiple data with single variable name which reduces the number of variables to be declared

2. Arrays are used to develop some algorithms like Bubble sort, Insertion sort, Selection sort etc

3. Arrays can be used to perform matrix operations

4. Arrays can be used for CPU scheduling

5. Array are used to implement data structure for example Stack, ArrayList, Queues etc

---

=> Point to remember for Array & Collection Framework :-

1. -> Array is java language feature inbuilt support provided by Sun Microsystems. We have to develop algorithms to sort or insert or delete etc

-> Collection Framework are API feature. It provides predefined classes and interfaces and methods by which we can easily iterate or delete or sort the elements

2. -> Array can store primitive (int, char etc) and non-primitive (objects) data types

-> Collection Framework can store only non-primitive data types (objects)

3. -> Array can store only homogeneous data types i.e. array can store only similar type of data

-> Collection Framework can store hetrogeneous data i.e. we can store different type of data

4. -> The size of an array cannot be increased or decreased according to our requirement at runtime

-> The size of collection can be increased or decreased according to our needs

5. -> Array are not good with respect to memory

-> Collection framework are very good with respect to memory

6. -> Arrays are good by performance wise

-> Collection are not good by performance wise

=> What is Collection Framework ?

-> Collection Framework consists of 2 words i.e. Collection and Framework

= Collection is a single entity or an object which contains multiple data

= Framework represents the library

-> Collection framework is the set of classes and interfaces that implement commonly reusable collection data structure

-> Collection framework contains 2 main parts :-

1. java.util.Collection
2. java.util.Map

-> "9 key interfaces" of Collection Framework

1. Collection
2. List
3. Set
4. SortedSet
5. NavigableSet
6. Queue

7. Map
8. SortedMap
9. NavigableMap

-> Hierarchy of Collection Framework :-

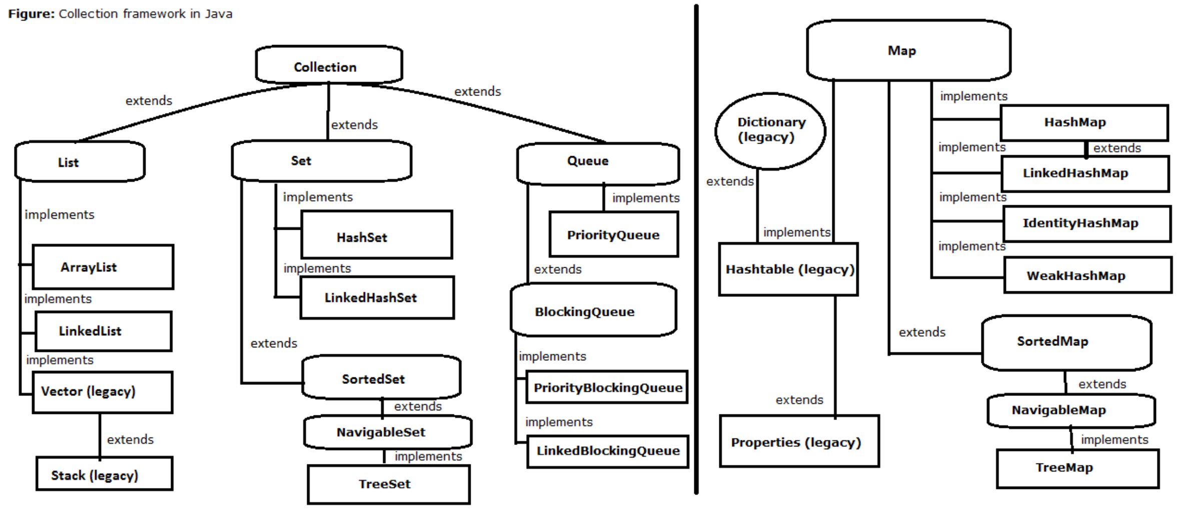


Figure: Collection framework in Java
Collection
extends
extends
extends
List
Set
Queue
implements
ArrayList
implements
LinkedList
implements
Vector (legacy)
extends
Stack (legacy)
implements
HashSet
implements
LinkedHashSet
extends
SortedSet
extends
NavigableSet
implements
TreeSet
implements
PriorityQueue
extends
BlockingQueue
implements
PriorityBlockingQueue
implements
LinkedBlockingQueue
Map
Dictionary (legacy)
extends
implements
Hashtable (legacy)
extends
Properties (legacy)
implements
HashMap
implements
extends
LinkedHashMap
implements
IdentityHashMap
implements
WeakHashMap
extends
SortedMap
extends
NavigableMap
implements
TreeMap

(image copyright to https://www.benchresources.net/collection-framework-in-java/)

-> In Collection, we can store the data directly but in Map we can store the data in key-value pairs

Company Links & Contacts

Company Name: Smart Programming (+91 62838-30308)

Address : Chandigarh & Mohali (Punjab), India

Websites: https://www.smartprogramming.in/

https://courses.smartprogramming.in

Android App:
https://play.google.com/store/apps/details?id=com.smartprogramming

YouTube Channel:
https://www.youtube.com/c/SmartProgramming

Smart Programming : YouTube Channel

An investment in Knowledge pays the best interest….

Smart Programming
We Educate - We Develop

☎ +91 62838-30308
Call us to Learn Latest Technologies

City : Mohali (Punjab), & Chandigarh (India)

WEBSITE : http://www.smartprogramming.in

BUY COURSES ON : https://courses.smartprogramming.in

YOUTUBE CHANNEL : Smart Programming (https://www.youtube.com/c/SmartProgramming)

ANDROID APP : Smart Programming (https://play.google.com/store/apps/details?id=com.smartprogramming)

https://www.facebook.com/smartprogramming.india

https://www.instagram.com/smart_programming

Java
android
python
Data Science
Machine Learning
Artificial Intelligence
M E A N
Full Stack
BIG DATA
hadoop

Collection Interface in Java

=> Collection :-

-> Collection is an interface which is present in java.util package

-> Syntax : public interface Collection<E> extends Iterable<E> { - }

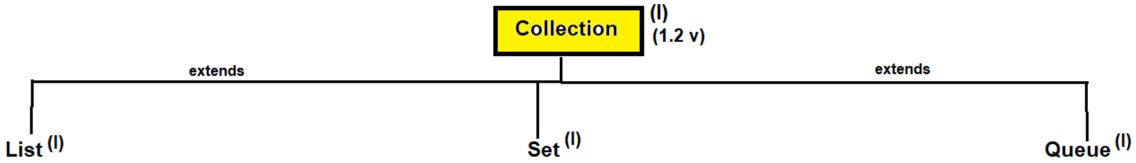
-> Collection was introduced in JDK 1.2 version

-> Collection is an object which is used to represent a group of individual objects as a single unit

-> Collection interface is the root interface of Collection Framework

-> There is no concrete class which implements the Collection interface directly but there are interfaces which inherit the Collection interface i.e. List, Set & Queue

-> Hierarchy of Collection interface :-


Collection
(I)
(1.2 v)
extends
extends
List (I)
Set (I)
Queue (I)

-> Collection interface contains most common methods which are applicable for any collection object

-> Methods of Collection Interface :-

1. boolean add(Object obj);
2. boolean addAll(Collection c);
3. boolean remove(Object obj);
4. boolean removeAll(Collection c);
5. default boolean removeIf(-) { - }
6. boolean retainAll(Collection c);

7. void clear();

8. boolean contains(Object obj);
9. boolean containsAll(Collection c);

10. boolean isEmpty();
11. int size();
12. Iterator iterator();
13. Object toArray();

14. boolean equals(Object obj);
15. int hashCode();

=> What is difference between Collection & Collections

1. -> Collection is an interface

   -> Collections is a utility class

2. -> Collection is an object which is used to represent a group of individual objects as a single unit

   -> Collections defines several utility methods that are used to operate on collection objects like sorting, searching etc

3. -> Collection interface contains default, abstract methods and static methods

   -> Collections class contains only static methods

=> What is Utility Class in Java ?

-> Utility class is also known as helper class which cannot be instantiated

-> Utility class contains only static methods

-> Examples are Arrays, Collections

-> How we can create utility class :-

1. declare the class as public and final
2. we have to declare private constructor to prevent object creation
3. class should contain only static methods and does not contain abstract methods
4. every method should have deep documentation

=> What is Utility Methods ?

-> Utility methods perform common, often reused methods.

-> Utility methods are always static type

-> Examples are sorting, searching, methods performing string manipulation, methods connecting to databases etc

Company Links & Contacts

Company Name: Smart Programming (+91 62838-30308)

Address : Chandigarh & Mohali (Punjab), India

Websites: https://www.smartprogramming.in/

https://courses.smartprogramming.in

Android App: https://play.google.com/store/apps/details?id=com.smartprogramming

YouTube Channel: https://www.youtube.com/c/SmartProgramming

Smart Programming : YouTube Channel

An investment in Knowledge pays the best interest….

Smart Programming
We Educate - We Develop

☎ +91 62838-30308
Call us to Learn
Latest Technologies

City : Mohali (Punjab),
& Chandigarh
(India)

WEBSITE : http://www.smartprogramming.in

BUY COURSES ON : https://courses.smartprogramming.in

YOUTUBE CHANNEL : Smart Programming (https://www.youtube.com/c/SmartProgramming)

ANDROID APP : Smart Programming
(https://play.google.com/store/apps/details?id=com.smartprogramming)

https://www.facebook.com/smartprogramming.india

https://www.instagram.com/smart_programming

Java
android
python
Data Science
Machine Learning
Artificial Intelligence
M E A N
Full Stack
BIG DATA
hadoop

List & ArrayList in Java

=> List Interface :-

- ➔ List is a interface which is present in java.util package
- ➔ List is the child interface of Collection interface
- ➔ Syntax : public interface List extends Collection { - }
- ➔ List was introduced in JDK 1.2 version
- ➔ Hierarchy of List interface :-

➔ Properties of List Interface :-

1. List is an index based Data Structure which means that first element will be inserted at 0 index position
2. List can store different data types or heterogeneous elements
3. We can store duplicate elements in the List
4. We can store any number of null values in the List
5. List follows the insertion order which means the sequence in which we are inserting the

elements, in the same sequence we can retrieve the elements

6. List does not follow the sorting order

➔ Methods of List Interface :-

1. List contains all the methods of Collection interface

2. void add(int index, Object obj);
3. boolean addAll(int index, Collection c);
4. Object get(int index);
5. Object remove(int index);
7. 6. Object set(int index, Object newobj); //set method is used to replace the object at given index position
8. 7. int indexOf(Object obj); //it will return the index position of provided object and if object is not found then it will return -1
9. int lastIndexOf(Object obj);

=> ArrayList :-

- ➔ ArrayList is an implemented class of List interface which is present in java.util package
- ➔ Syntax : public class ArrayList extends AbstractList implements List, RandomAccess, Cloneable, Serializable
- ➔ The underline Data-Structure of ArrayList is resizable array or growable array
- ➔ ArrayList was introduced in JDK 1.2 version

➔ Properties of ArrayList :-

1. ArrayList is an index based Data Structure which means that first element will be inserted at 0 index position
2. ArrayList can store different data types elements or hetrogeneous elements
3. We can store duplicate elements in the ArrayList
4. We can store any number of null values in the ArrayList

5. ArrayList follows the insertion order which means the sequence in which we are inserting the elements, in the same sequence we can retrieve the elements
6. ArrayList does not follow the sorting order (above properties are same as List interface)

7. ArrayList is non-synchronized collection because ArrayList does not contain any synchronized method
8. ArrayList allows more than one thread at one time
9. ArrayList allows parallel execution
10. ArrayList reduces the execution time which in turn makes the application fast
11. ArrayList is not threadsafe
12. ArrayList does not gurantee for data consistency

➔ Working of an ArrayList :-

1. When we create default ArrayList, a new ArrayList with initial capacity 10 is created (but size is 0)
2. When the ArrayList capacity is full, a new ArrayList will be created with new capacity. The new Capacity is calculated by this formula:-
    a. (CurrentCapacity * 3 / 2) + 1
3. Then all the elements will be copied into the new ArrayList (and due this this reason performance of an ArrayList decreases)
4. When new ArrayList is created automatically, then reference variable will point to the new ArrayList
5. Then old ArrayList object will be not referenced by any reference and then garbage collection will delete that object

Note : There is no way by which we can find the capacity of an ArrayList

➔ Contructors of ArrayList :-

1. ArrayList al=new ArrayList();
    a.= In this arraylist, an ArrayList collection object is created whose capacity is 10
2. ArrayList al=new ArrayList(int initialCapacity);
    a.= In this arraylist, an ArrayList object is created with provided initialCapacity
3. ArrayList al=new ArrayList(Collection c);
    a.= In this arraylist, another collection object is copied into new arraylist object

➔ When we should use ArrayList ?

= When we use retrival operation mostly (Retrival operation is fast in case of ArrayList because it implements RandomAccess interface)

➔ When we should not use ArrayList ?

= When we have mostly insertion or deletion operation, then we should not use ArrayList

=> RandomAccess interface :-

-> RandomAccess interface is a marker interface that means it does not contain any methods or fields (variables)

-> The purpose of RandomAccess interface is to retrieve any random element in collection object at the same speed. For example we have collection object having 1 crore elements, we have to search 3rd element or middle element or last element then it will search with the same speed

-> There are only 2 classes which inherits the RandomAccess interface

1. ArrayList
2. Vector

=> Cloneable Interface :-

-> Cloneable interface is also a marker interface

-> It was introduced in JDK 1.0 version

-> It is used to clone the object without using the new keyword

Company Links & Contacts

Company Name: Smart Programming (+91 62838-30308)

Address : Chandigarh & Mohali (Punjab), India

Websites: https://www.smartprogramming.in/

https://courses.smartprogramming.in

Android App:
https://play.google.com/store/apps/details?id=com.smartprogramming

YouTube Channel:
https://www.youtube.com/c/SmartProgramming

Smart Programming : YouTube Channel

An investment in Knowledge pays the best interest….

Smart Programming

We Educate - We Develop

☎ +91 62838-30308

Call us to Learn Latest Technologies

City : Mohali (Punjab), & Chandigarh (India)

WEBSITE : http://www.smartprogramming.in

BUY COURSES ON : https://courses.smartprogramming.in

YOUTUBE CHANNEL : Smart Programming (https://www.youtube.com/c/SmartProgramming)

ANDROID APP : Smart Programming (https://play.google.com/store/apps/details?id=com.smartprogramming)

https://www.facebook.com/smartprogramming.india

https://www.instagram.com/smart_programming

Java
android
python
Data Science
Machine Learning
Artificial Intelligence
M E A N
Full Stack
BIG DATA
hadoop

LinkedList in Java

=> LinkedList :-

- ➔ LinkedList is an implementation class of List interface which is present in java.util package
- ➔ Syntax : public class LinkedList extends AbstractSequentialList implements List, Deque, Cloneable, Serializable { - }
- ➔ The underline data structure of LinkedList is Double Linked List or Circular Linked List
- ➔ LinkedList was introduced in JDK 1.2 version

➔ Properties of LinkedList :-

1. LinkedList is an index based Data Structure which means that first element will be inserted at 0 index position
2. LinkedList can store different data types or hetrogeneous elements
3. We can store duplicate elements in the LinkedList
4. We can store any number of null values in the LinkedList

5. LinkedList follows the insertion order which means the sequence in which we are inserting the elements, in the same sequence we can retrieve the elements
6. LinkedList does not follow the sorting order

(above properties are same as List interface)

7. LinkedList is non-synchronized collection because LinkedList does not contain any synchronized method
8. LinkedList allows more than one thread at one time
9. LinkedList allows parallel execution
10. LinkedList reduces the execution time which in turn makes the application fast
11. LinkedList is not threadsafe
12. LinkedList does not gurantee for data consistency

-> Working of LinkedList :-

1. Types of LinkedList (all programming languages) :-

    a. Single Linked List

    b. Double Linked List

    c. Circular Linked List

2. Linked List are linear data structure in which elements are not stored in contigeous memory locations.

3. There is no capacity concept in LinkedList like ArrayList

-> Constructors :-

1. public LinkedList()

2. public LinkedList(Collection c)

-> Methods of LinkedList :-

1. Methods of Collection Interface
2. Methods of List Interface

3. public void addFirst(Object obj)
4. public void addLast(Object obj)

5. public Object getFirst()
6. public Object getLast()

7. public Object removeFirst()
8. public Object removeLast()

-> When we should use LinkedList ?

= LinkedList is best when we have to use insertion or deletion operations

-> When we should not use LinkedList ?

= LinkedList is worst in case of retrieval or searching operations (as LinkedList does not inherit RandomAccess interface)

-> What is difference between ArrayList & LinkedList ?

1. ArrayList underline data structure is dynamic array or resizable array

LinkedList underline data structure is double linked list or circular linked list

2. ArrayList stores the elements in contigeous memory locations

LinkedList does not store the elements in contigeous memory locations

3. ArrayList acts as List

LinkedList can acts as List or Deque

4. ArrayList is good in case of retrieval operations

LinkedList is good in case of insertion and deletion operations

5. ArrayList is worst in case of insertion or deletion operations

LinkedList is worst in case of retrieval operations

Company Links & Contacts

Company Name: Smart Programming (+91 62838-30308)

Address : Chandigarh & Mohali (Punjab), India

Websites: https://www.smartprogramming.in/
https://courses.smartprogramming.in

Android App:
https://play.google.com/store/apps/details?id=com.smartprogramming

YouTube Channel:
https://www.youtube.com/c/SmartProgramming

Smart Programming : YouTube Channel

An investment in Knowledge pays the best interest….

C
C++
C#
Java
ANDROID
python
Data Science
Machine Learning
Artificial Intelligence
M E A N
Full Stack
BIG DATA
hadoop

Smart Programming
We Educate - We Develop

☎ +91 62838-30308
Call us to Learn Latest Technologies

City : Mohali (Punjab), & Chandigarh (India)

WEBSITE : http://www.smartprogramming.in

BUY COURSES ON : https://courses.smartprogramming.in

YOUTUBE CHANNEL : Smart Programming (https://www.youtube.com/c/SmartProgramming)

ANDROID APP : Smart Programming (https://play.google.com/store/apps/details?id=com.smartprogramming)

https://www.facebook.com/smartprogramming.india

https://www.instagram.com/smart_programming

Vector in Java

=> Vector :-

- ➔ Vector is an implementation class of List interface which is present in java.util package
- ➔ Syntax : public class Vector extends AbstractList implements List, RandomAccess, Cloneable, Serializable { - }
- ➔ The underline data structure of Vector is resizable array or growable array
- ➔ Vector was introduced in JDK 1.0 version
- ➔ Vector class is also known as legacy class. (Legacy class is the class which was formed in previous version and was restructured or re-engineered in new version)

-> Properties of Vector :-

1. Vector is an index based Data Structure which means that first element will be inserted at 0 index position
2. Vector can store different data types elements or hetrogeneous elements
3. We can store duplicate elements in the Vector

4. We can store any number of null values in the Vector
5. Vector follows the insertion order which means the sequence in which we are inserting the elements, in the same sequence we can retrieve the elements
6. Vector does not follow the sorting order (above properties are same as List interface)

7. Vector is synchronized collection because Vector contains many synchronized method
8. Vector does not allow more than one thread at one time
9. Vector does not allow parallel execution or Vector allows sequential execution
10. Vector increases the execution time which in turn makes the application slow
11. Vector is threadsafe
12. Vector gurantee for data consistency

-> Working of Vector :-

1. When we create a vector, a vector of 10 initial capacity is created
2. When the vector is full, then new vector will be created automatically with new capacity = current capacity * 2;
3. When new vector is created then all the elements from old vector will copied to new vector and then the reference variable will point to the new vector and garbage collector will delete the previous vector from the memory

Note : In vector we can find the capacity

-> Constructors of Vector :-

1. public Vector()

2. public Vector(int capacity)

3. public Vector(int capacity, int incremental_ratio)

4. public Vector(Collection c)

-> Methods :-

1. Vector contains all the methods of Collection interface

2. Vector contains all the methods of List interface

1. public synchronized int capacity()

2. public synchronized void addElement(Object obj)

3. public synchronized Object firstElement()

4. public synchronized Object lastElement()

5. public synchronized boolean removeElement(Object obj)

6. public synchronized void removeElementAt(int index)

7. public synchronized void removeAllElements()

-> When we should use Vector ?

= We should use Vector in case of retrieval or searching operations

(Vector inherits the RandomAccess interface)

-> When we should not use Vector ?

= We should not use Vector in case of insertion or deletion of elements

-> What is difference between ArrayList & Vector :-

1. ArrayList was introduced in JDK 1.2 version

   Vector was introduced in JDK 1.0 version

2. ArrayList is not a legacy class

   Vector is legacy class

3. ArrayList is non-synchronized collection

Vector is synchronized collection

4. ArrayList allows more than one thread at one time

Vector does not allow more than one thread at one time

5. ArrayList allows the parallel execution

Vector does not allow parallel execution

6. ArrayList decreases the execution time which in turn makes the application fast

Vector increases the execution time which in turn makes the application slow

7. ArrayList is not threadsafe

Vector is threadsafe

8. ArrayList does not gurantee for data consistency

Vector gurantee for data consistency

9. In case of ArrayList new capacity = (present capacity * 3/2)+1

In case of Vector new capacity = present capacity * 2

10. In ArrayList we cannot find the capacity

In Vector we can find the capacity

Company Links & Contacts

Company Name: Smart Programming (+91 62838-30308)

Address : Chandigarh & Mohali (Punjab), India

Websites: https://www.smartprogramming.in/

https://courses.smartprogramming.in

Android App: https://play.google.com/store/apps/details?id=com.smartprogramming

YouTube Channel: https://www.youtube.com/c/SmartProgramming

Smart Programming : YouTube Channel

An investment in Knowledge pays the best interest….

Smart Programming
We Educate - We Develop

☎ +91 62838-30308
Call us to Learn
Latest Technologies

City : Mohali (Punjab),
& Chandigarh
(India)

WEBSITE : http://www.smartprogramming.in

BUY COURSES ON : https://courses.smartprogramming.in

YOUTUBE CHANNEL : Smart Programming (https://www.youtube.com/c/SmartProgramming)

ANDROID APP : Smart Programming
(https://play.google.com/store/apps/details?id=com.smartprogramming)

https://www.facebook.com/smartprogramming.india

https://www.instagram.com/smart_programming

Java
android
python
Data Science
Machine Learning
Artificial Intelligence
M E A N
Full Stack
BIG DATA
hadoop

Stack in Java

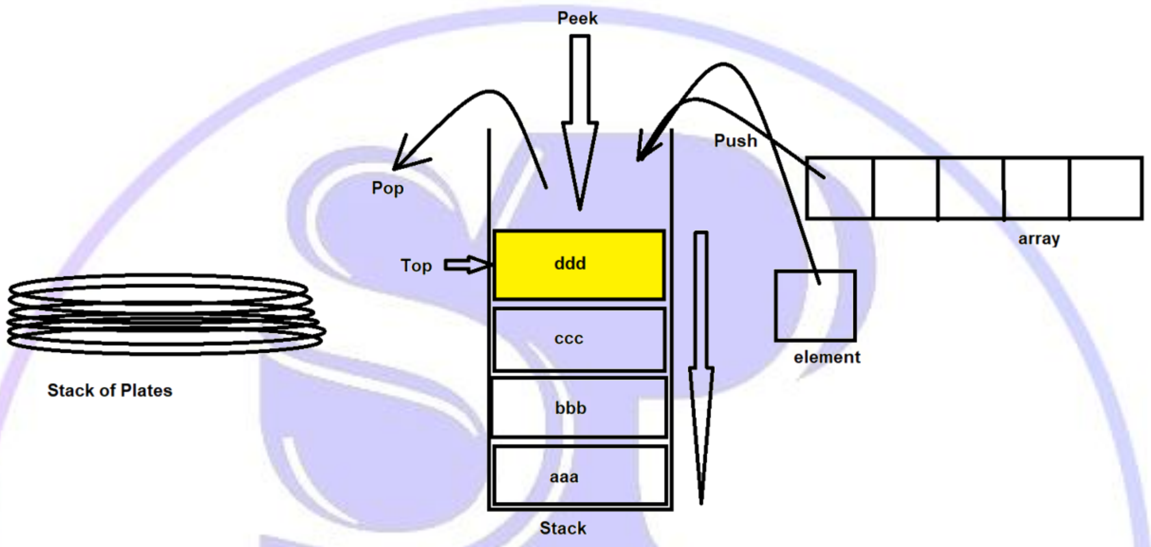
=> Stack :-

- ➔ Stack is the child class of Vector class present in java.util package
- ➔ Syntax : public class Stack extends Vector { - }
- ➔ Stack was introduced in JDK 1.0 version
- ➔ Stack is also known as legacy class
  (legacy class is the class which was re-engineered or restrctured in newer version)

-> Properties of Stack :-

1. Stack class was specially designed for Last In First Out (LIFO) but we can change this algorithm according to our requirement
2. Stack can be implemented using Array, ArrayList, LinkedList or Vector
3. Stack is also index based data structure

-> Working of Stack :-


Peek
Pop
Push
Top
ddd
ccc
bbb
aaa
Stack
array
element
Stack of Plates

In java stack is implemented by LIFO (Last In First Out) algorithm

-> Constructors :-

1. public Stack() { - }

-> Methods :-

1. Methods of Vector, List and Collections

2. public Object push(Object obj)
3. public synchronized Object pop()
4. public synchronized Object peek()
5. public synchronized int search(Object obj)
6. public boolean empty()

Company Links & Contacts

Company Name: Smart Programming (+91 62838-30308)

Address : Chandigarh & Mohali (Punjab), India

Websites: https://www.smartprogramming.in/

https://courses.smartprogramming.in

Android App: https://play.google.com/store/apps/details?id=com.smartprogramming

YouTube Channel: https://www.youtube.com/c/SmartProgramming

Smart Programming : YouTube Channel

An investment in Knowledge pays the best interest....

Smart Programming
We Educate - We Develop
+91 62838-30308
Call us to Learn Latest Technologies
City : Mohali (Punjab), & Chandigarh (India)
WEBSITE : http://www.smartprogramming.in
BUY COURSES ON : https://courses.smartprogramming.in
YOUTUBE CHANNEL : Smart Programming (https://www.youtube.com/c/SmartProgramming)
ANDROID APP : Smart Programming (https://play.google.com/store/apps/details?id=com.smartprogramming)
https://www.facebook.com/smartprogramming.india
https://www.instagram.com/smart_programming
Java
android
python
Data Science
Machine Learning
Artificial Intelligence
M E A N
Full Stack
BIG DATA
hadoop

Cursors in Java

=> Cursors :-

- ➔ In java, whenever we print the object reference, internally JVM will call toString() method of Object class. In case of simple object it will print ClassName@referencevalue but in case of printing Collection object it will print the elements present in Collection object.

- ➔ When we print the collection object it will retrieve all the elements at one time but we want to retrieve the elements one by one then we have to use cursors

-> Types of cursors :-

1. Enumeration
2. Iterator
3. ListIterator

=> Enumeration :-

- ➔ Enumeration is the cursor which is used to get the elements one by one from the collection object
- ➔ Enumeration was introduced in JDK 1.0 version
- ➔ Enumeration is used only for legacy class

-> Steps "how to use" enumeration cursor :-

1. Create Enumeration Cursor Object

-> public Enumeration elements()

(this method is present in Vector & Stack legacy class)

2. Read one by one all the elements from Enumeration Cursor

-> public boolean hasMoreElements()

-> public Object nextElement()

(these methods are present in Enumeration interface)

-> Limitations of Enumeration :-

1. It can be used only with Legacy class and thus it is not universal cursor
2. By using enumeration cursor we can only perform read operation but not update or remove operation
3. It can be used to traverse the elements only in forward direction

---

=> Iterator :-

- ➔ Iterator is a cursor which is used to get the elements one by one from the collection object
- ➔ It is universal cursor which means that we can use it with all collection objects
- ➔ It can be used for read and remove operation
- ➔ It was introduced in JDK 1.2 version

-> Steps "how to use" iterator cursor :-

1. Create Iterator cursor object :-

   -> public Iterator iterator()

   (this method is present for every collection object)

2. Read one by one all the elements from iterator cursor :-

   -> public boolean hasNext()

   -> public Object next()

   -> public void remove()

   (these methods are present in Iterator interface)

-> Limitations of Iterator cursor :-

1. It can be used only for read and remove operation but not for replacement or addition operation

2. It can be used to iterate the elements only in forward direction

---

=> ListIterator :-

- ➔ ListIterator is a cursor which is used to get the elements one by one from collection object
- ➔ ListIterator is bi-directional cursor which means it can be used to traverse the elements in forward or backword direction
- ➔ It can be used to read, remove, insert and replace operations
- ➔ It was introduced in JDK 1.2 version

-> Steps "how to use" ListIterator cursor :-

1. Create ListIterator cursor object :-

-> public ListIterator listIterator();

(which is present in only List implementation classes)

2. Read one by one all the elements from ListIterator cursor :-

-> public boolean hasNext()

-> public Object next()

-> public int nextIndex()

(above methods are used to traverse the elements in forward direction)

-> public boolean hasPrevious()

-> public Object previous()

-> public int previousIndex()

(above methods are used to traverse the elements in backword direction)

-> public void remove()

-> public void add(Object obj)

-> public void set(Object obj)

(above methods are used to remove, add and replace operations)

-> Limitations of ListIterator cursor :-

1. It can be used only with List implemented classes thus it is not universal cursor

=> What is difference between Enumeration, Iterator & ListIterator :-

1. Enumeration can be used only with legacy classes

   Iterator can be used for any collection object

   ListIterator can be used only for List implemented classes

2. Enumeration can be used to traverse the elements only in forward direction

   Iterator can be used to traverse the elements only in forward direction

   ListIterator can be used to traverse the elements in forward and backword direction

3. Enumeration is used only for read operation

   Iterator can be used for read and remove operation

ListIterator can be used for read, remove, add and replace operations

Company Links & Contacts

Company Name: Smart Programming (+91 62838-30308)

Address : Chandigarh & Mohali (Punjab), India

Websites: https://www.smartprogramming.in/
https://courses.smartprogramming.in

Android App:
https://play.google.com/store/apps/details?id=com.smartprogramming

YouTube Channel:
https://www.youtube.com/c/SmartProgramming

Smart Programming : YouTube Channel

An investment in Knowledge pays the best interest....

Smart Programming
We Educate - We Develop

+91 62838-30308
Call us to Learn Latest Technologies

City : Mohali (Punjab), & Chandigarh (India)

WEBSITE : http://www.smartprogramming.in

BUY COURSES ON : https://courses.smartprogramming.in

YOUTUBE CHANNEL : Smart Programming (https://www.youtube.com/c/SmartProgramming)

ANDROID APP : Smart Programming (https://play.google.com/store/apps/details?id=com.smartprogramming)

https://www.facebook.com/smartprogramming.india

https://www.instagram.com/smart_programming

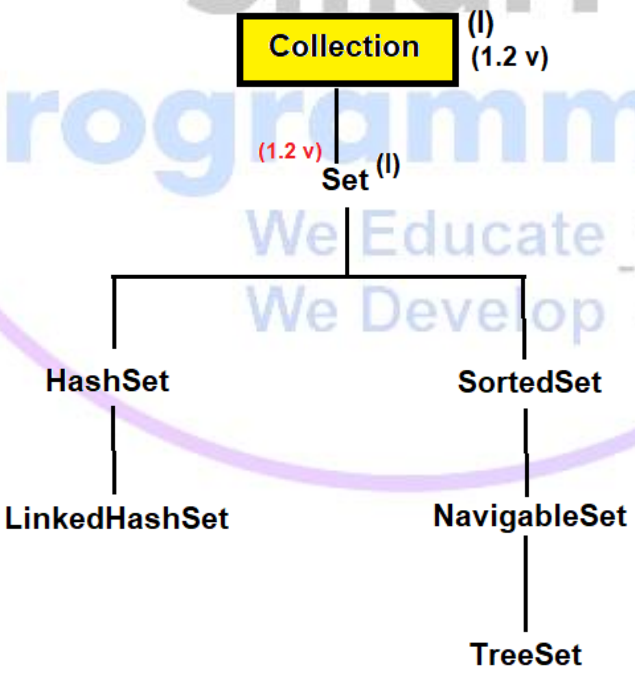
Java
android
python
Data Science
Machine Learning
Artificial Intelligence
M E A N
Full Stack
BIG DATA
hadoop

Set & HashSet in Java

=> Set :-

- Set is an interface which is present in java.util package
- Set is the child interface of Collection interface
- Syntax : public interface Set extends Collection { - }
- Set was introduced in JDK 1.2 version

- Hierarchy of Set interface :-


Collection
(I)
(1.2 v)
(1.2 v)
Set (I)
HashSet
SortedSet
LinkedHashSet
NavigableSet
TreeSet

-> Properties of Set interface :-

1. Set is not an index based data structure, it stores the elements as per elements "hashcode" values
2. Set does not follow the insertion order (except LinkedHashSet)
3. Set does not follow the sorting order (except SortedSet, NavigableSet & TreeSet)
4. Set can store different data types or hetrogeneous elements (except SortedSet, NavigableSet, TreeSet)
5. We cannot store duplicate elements in Set
6. We can store only one null value in Set

-> Methods of Set interface :-

= approx same methods as that of Collection interface

=> Difference between List & Set :-

1. List is index based data structure which means in list data is stored by using index position

Set is not an index based data structure, it stores the data according to hashcode values of the elements

2. List allows duplicate elements

Set does not allow duplicate elements

3. List can store any number of null values

Set can store only one null value

4. List follows the insertion order

Set does not follows the insertion order

5. In case of List we can use Iterator & ListIterator cursor

In case of Set we can use only Iterator cursor

6. List is used in case of retrieveing the elements

Set is used when we does not want to allow duplicacy

=> HashSet :-

- ➔ HashSet is an implemented class of Set interface which is present in java.util package
- ➔ Syntax : public class HashSet extends AbstractSet implements Set, Cloneable, Serializable { - }

- ➔ The underline data structure of HashSet is Hashtable (HashSet is backed up by Map)
- ➔ HashSet was introduced in JDK 1.2 version

- ➔ Properties of HashSet :-

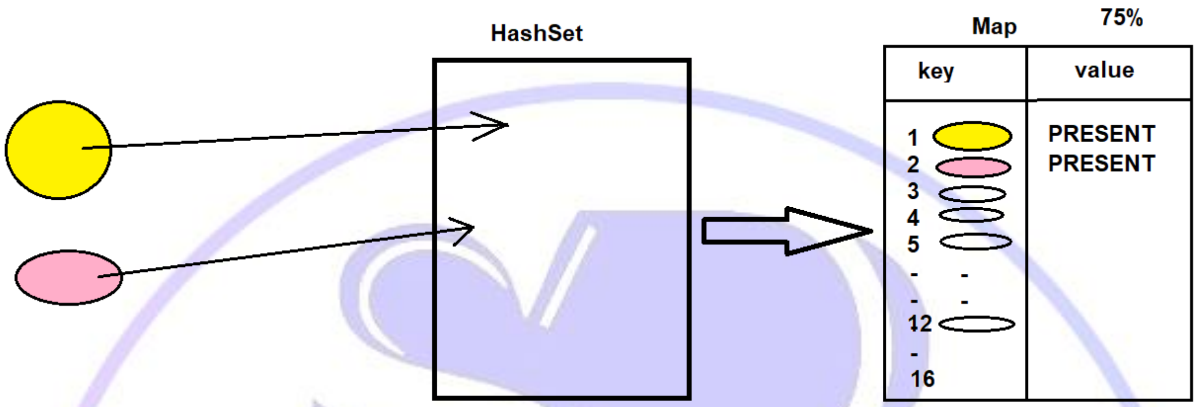
1. HashSet is not an index based data structure, it stores the elements according to elements hashcode values
2. HashSet can store different data types of hetrogeneous elements
3. HashSet cannot store the duplicate elements
4. HashSet can store maximum only one null value
5. HashSet does not follows the insertion order
6. HashSet does not follow the sorting order
    a. (same properties as Set interface)

7. HashSet is non-synchronized collection because HashSet does not contain any synchronized methods

8. HashSet allows more than one thread at one time
9. HashSet allows the parallel execution
10. HashSet reduces the execution time which in turn makes our application fast
11. HashSet is not threadsafe
12. HashSet does not gurantee for data consistency

-> Working of HashSet :-

1. HashSet internally works on the basis of Hashtable (it internally backed up by Map.)
2. When we insert any element in HashSet, it stores as a key inside the Map and at value position PRESENT reference variable is stored which is the reference variable of Object class
3. Initial capacity of HashSet is 16 elements
4. Its load factor or fill ratio is 75%
5. After load is filled then a new Map is created with its double capacity


HashSet
Map
75%
key
value
1
2
3
4
5
-
-
12
-
16
PRESENT
PRESENT

-> Constructors of HashSet :-

1. public HashSet()

2. public HashSet(Object obj)

3. public HashSet(int initialCapacity)

4. public HashSet(int initialCapacity, float loadFactor)

-> Methods of HashSet :-

= approx same methods as that of Collection or Set interface

-> When we should use HashSet ?

= HashSet is good for searching or retriving operations

Company Links & Contacts

Company Name: Smart Programming (+91 62838-30308)

Address : Chandigarh & Mohali (Punjab), India

Websites: https://www.smartprogramming.in/

https://courses.smartprogramming.in

Android App: https://play.google.com/store/apps/details?id=com.smartprogramming

YouTube Channel: https://www.youtube.com/c/SmartProgramming

Smart Programming : YouTube Channel

An investment in Knowledge pays the best interest....

Smart Programming
We Educate - We Develop

+91 62838-30308
Call us to Learn
Latest Technologies

City : Mohali (Punjab),
& Chandigarh
(India)

WEBSITE : http://www.smartprogramming.in

BUY COURSES ON : https://courses.smartprogramming.in

YOUTUBE CHANNEL : Smart Programming (https://www.youtube.com/c/SmartProgramming)

ANDROID APP : Smart Programming
(https://play.google.com/store/apps/details?id=com.smartprogramming)

https://www.facebook.com/smartprogramming.india

https://www.instagram.com/smart_programming

Java
android
python
Data Science
Machine Learning
Artificial Intelligence
M E A N
Full Stack
BIG DATA
hadoop

LinkedHashSet, SortedSet, NavigableSet in Java

=> LinkedHashSet :-

- LinkedHashSet is the child class of HashSet which is present in java.util package
- Syntax : public class LinkedHashSet extends HashSet implements Set, Cloneable, Serializable { - }
- The underline data structure of LinkedHashSet is "Hashtable + LinkedList"
- LinkedHashSet was introduced in JDK 1.4 version

- Properties of LinkedHashSet :-

= All properties of LinkedHashSet is same as HashSet except LinkedHashSet follows the insertion order

- Constructors of LinkedHashSet :-

= same constructors as HashSet

➔ Methods of LinkedHashSet :-

= Same methods as HashSet

➔ When we should use LinkedHashSet :-

= If we have to create cache based applications then we can use LinkedHashSet

➔ Difference between HashSet & LinkedHashSet :-

1. HashSet was introduced in JDK 1.2 version

LinkedHashSet was introduced in J2SE 1.4 version

2. HashSet does not follow the insertion order

LinkedHashSet follows the insertion order

3. HashSet underline data structure is "Hashtable"

LinkedHashSet underline data structure is "Hashtable + LinkedList"

=> SortedSet :-

- ➔ SortedSet is a child interface of Set interface which is present in java.util package
- ➔ Syntax : public interface SortedSet extends Set { - }
- ➔ SortedSet was introduced in JDK 1.2 version

➔ Properties of SortedSet :-

1. SortedSet is not an index based data structure
2. SortedSet does not follows the insertion order
3. SortedSet follows the sorting order
4. SortedSet can store same data types or homogeneous elements. If we provide different data type element if will provide exception i.e. java.lang.ClassCastException
5. SortedSet cannot store the duplicate elements
6. We should not store null value in SortedSet because SortedSet follows the sorting order so while comparing the elements with null value, it will provide the java.lang.NullPointerException
7. SortedSet allows Comparable objects bydefault, but if we insert non-comparable objects, then it will provide an exception i.e. java.lang.ClassCastException

8. SortedSet is non-synchronized collection

➔ Methods of SortedSet :-

1. Object first();
2. Object last();
3. SortedSet headSet(Object toElement);
4. SortedSet tailSet(Object fromElement);
5. SortedSet subSet(Object fromElement, Object toElement);

=> NavigableSet :-

➔ NavigableSet is the child interface of SortedSet interface which is present in java.util package

➔ Syntax : public interface NavigableSet extends SortedSet { - }

➔ NavigableSet was introduced in Java SE 6 version

➔ Properties of NavigableSet :-

= NavigableSet has the same properties as that of SortedSet but it provides some extra navigable methods

➔ Methods :

1. public NavigableSet descendingSet()
2. public Object ceiling(Object obj)
3. public Object higher(Object obj)
4. public Object floor(Object obj)
5. public Object lower(Object obj)
6. public Object pollFirst()
7. public Object pollLast()

Company Links & Contacts

Company Name: Smart Programming (+91 62838-30308)

Address : Chandigarh & Mohali (Punjab), India

Websites: https://www.smartprogramming.in/
https://courses.smartprogramming.in

Android App:
https://play.google.com/store/apps/details?id=com.smartprogramming

YouTube Channel:
https://www.youtube.com/c/SmartProgramming

Smart Programming : YouTube Channel

An investment in Knowledge pays the best interest….

Smart Programming
We Educate - We Develop

+91 62838-30308
Call us to Learn
Latest Technologies

City : Mohali (Punjab),
& Chandigarh
(India)

WEBSITE : http://www.smartprogramming.in

BUY COURSES ON : https://courses.smartprogramming.in

YOUTUBE CHANNEL : Smart Programming (https://www.youtube.com/c/SmartProgramming)

ANDROID APP : Smart Programming
(https://play.google.com/store/apps/details?id=com.smartprogramming)

https://www.facebook.com/smartprogramming.india

https://www.instagram.com/smart_programming

Java
android
python
Data Science
Machine Learning
Artificial Intelligence
M E A N
Full Stack
BIG DATA
hadoop

TreeSet in Java

=> TreeSet :-

- ➔ TreeSet is the direct implementation for NavigableSet interface but it also provides the implementation for SortedSet, Set & Collection interface
- ➔ Syntax : public class TreeSet extends AbstractSet implements NavigableSet, Cloneable, Serializable { - }
- ➔ The underline data structure of TreeSet is "Balanced Tree"
- ➔ TreeSet was introduced in JDK 1.2 version

➔ **Properties of TreeSet :-**

1. TreeSet is not index based data structure
2. TreeSet does not follows the insertion order
3. TreeSet follows the sorting order, it may be default sorting order or customized sorting order
4. TreeSet stores the same data type elements or homogeneous elements. If we provide different data type elements in TreeSet then it will provide java.lang.ClassCastException

5. TreeSet cannot stores the duplicate elements
6. TreeSet can store only one null value.

7. TreeSet is non-synchronized collection
8. TreeSet allows more than one thread at one time
9. TreeSet allows the parallel execution
10. TreeSet reduces the execution time which makes our application fast
11. TreeSet is not threadsafe
12. TreeSet does not provide gurantee for data consistency

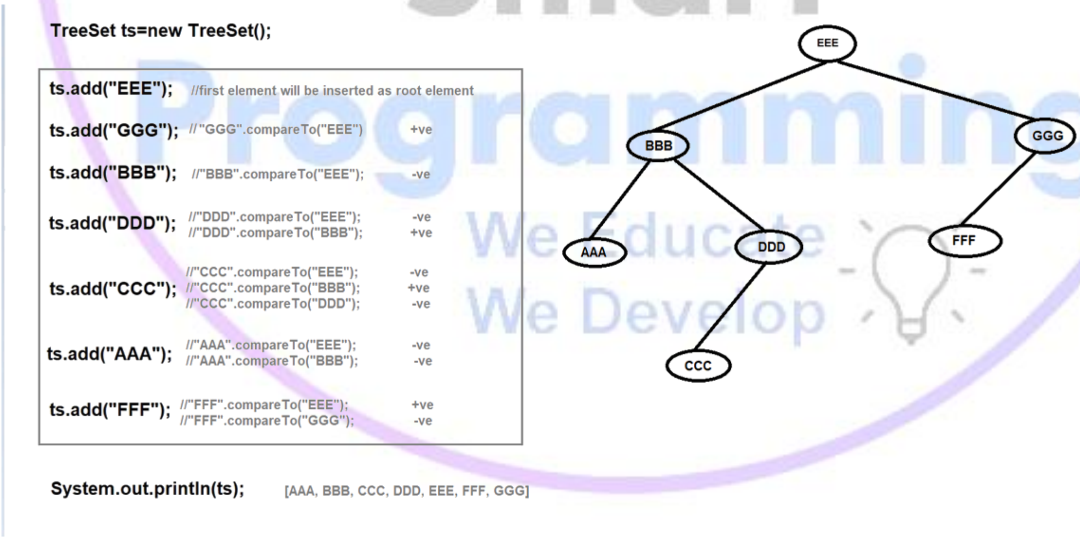
➔ Working of TreeSet :-

1. When we insert the first element, it will become the root node.
2. Then when we insert next element, it will compare with the root node ("next element".compareTo("root element");), if the return value is -ve, then it will goes to the left

side and if the return value is +ve then it will goes to the right side

3. Then again it will check wether there is any parent node or not, if there is any parent node, then again it will compare and check the return value, then if return value is +ve, it will goes to right side and if return value is -ve, it will goes to the left side

4. When we retrieve the elements, it will retrieve as "LEFT - ROOT - RIGHT"



```
TreeSet ts=new TreeSet();

ts.add("EEE");   //first element will be inserted as root element

ts.add("GGG");  // "GGG".compareTo("EEE")          +ve

ts.add("BBB");   //"BBB".compareTo("EEE");         -ve

ts.add("DDD");  //"DDD".compareTo("EEE");          -ve
                //"DDD".compareTo("BBB");          +ve

                //"CCC".compareTo("EEE");          -ve
ts.add("CCC");  //"CCC".compareTo("BBB");          +ve
                //"CCC".compareTo("DDD");          -ve

ts.add("AAA");  //"AAA".compareTo("EEE");          -ve
                //"AAA".compareTo("BBB");          -ve

ts.add("FFF");  //"FFF".compareTo("EEE");          +ve
                //"FFF".compareTo("GGG");          -ve

System.out.println(ts);      [AAA, BBB, CCC, DDD, EEE, FFF, GGG]
```

➔ Constructors :

1. public TreeSet() //it will create an empty TreeSet object where the elements are inserted according to natural default sorting order
2. 2. public TreeSet(Comparator comparator) // it will create an empty TreeSet object where the elements will be inserted according to the customized sorting order
3. public TreeSet(Collection c) // we can pass any other collection object

4. public TreeSet(SortedSet s)

-> Methods of TreeSet :-

= contains all the methods of Collection, Set, SortedSet & NavigableSet interfaces

-> When we should use TreeSet :-

= When we want to store the large number of elements in sorting order and due to this retrival operation is fast

=> Cases for "null" value insertion in TreeSet :-

1. We can store only one null value

2. We can insert null value only at first position, but then if we insert any other element then it will provide "java.lang.NullPointerException"

3. NOTE : Until 1.6 version we can successfully insert the null value at first position but after 1.6 version we cannot store the null value even at first position

=> What is difference between HashSet, LinkedHashSet & TreeSet :-

1. HashSet underline data structure is "Hashtable"

LinkedHashSet underline data structure is "Hashtable + Linked List"

TreeSet underline data structure is "Balanced Tree"

2. HashSet does not allow insertion order

LinkedHashSet allow the insertion order

TreeSet does not allow the insertion order

3. HashSet does not allow the sorting order

LinkedHashSet does not allow sorting order

TreeSet allows the sorting order

4. HashSet allows the hetrogeneous objects

LinkedHashSet allows the hetrogeneous objects

TreeSet does not allow the hetrogeneous objects

5. HashSet allows the null value insertion

LinkedHashSet allows the null value insertion

TreeSet allows the null value but at first position (but this is applicable till java 1.6 version)

Company Links & Contacts

Company Name: Smart Programming (+91 62838-30308)

Address : Chandigarh & Mohali (Punjab), India

Websites: https://www.smartprogramming.in/

https://courses.smartprogramming.in

Android App:
https://play.google.com/store/apps/details?id=com.smartprogramming

YouTube Channel:
https://www.youtube.com/c/SmartProgramming

Smart Programming : YouTube Channel

An investment in Knowledge pays the best interest....

Smart Programming
We Educate - We Develop

☎ +91 62838-30308
Call us to Learn
Latest Technologies

City : Mohali (Punjab),
& Chandigarh
(India)

WEBSITE : http://www.smartprogramming.in

BUY COURSES ON : https://courses.smartprogramming.in

YOUTUBE CHANNEL : Smart Programming (https://www.youtube.com/c/SmartProgramming)

ANDROID APP : Smart Programming
(https://play.google.com/store/apps/details?id=com.smartprogramming)

https://www.facebook.com/smartprogramming.india

https://www.instagram.com/smart_programming

Java
android
python
Data Science
Machine Learning
Artificial Intelligence
M E A N
Full Stack
BIG DATA
hadoop

Comparable & Comparator in Java

=> Comparable :

- ➔ Comparable is an interface which is present in java.lang package
- ➔ It contains only one method i.e. compareTo()
  - ○ Prototype : public int compareTo(Object obj)
    - obj1.compareTo(obj2);
    - +ve - if obj1 is greater than obj2
    - -ve - if obj2 is greater than obj1
    - 0 - if obj1 is equals to obj2
- ➔ String and all wrapper classes (Integer, Long, Float etc) implements Comparable interface

- ➔ Problems with Comparable interface :-

1. By implementing Comparable interface, the properties of Original class will get changed
2. By this way we can sort only for one entity for example we can sort the student object either with name or rollno at one time

(To remove above problems java provided one interface i.e. Comparator)

=> Comparator :-

-> Comparator is an interface which is present in java.util package

-> Comparator interface contains 2 methods :-

1. public int compare(Object obj1, Object obj2)
2. public boolean equals(Object obj)

=> TASK : WAP to compare and add different elements according to length and alphabetical order

For example :

A, B, AA, BB, AAA, BBB, AAAA

=> What is difference between Comparable & Comparator interface :-

1. Comparable interface contains compareTo() method

Comparator interface contains compare() method

2. Comparable interface is present in java.lang package

Comparator interface is present in java.util package

3. By using Comparable interface original class properties will get changed

By using Comparator interface original class properties will not be changed

4. Comparable interface can sort only one entity

   Comparator interface can sort multiple entities

5. Comparable interface is used to implicit sorting

   Comparator interface is used for explicit sorting

Company Links & Contacts

Company Name: Smart Programming (+91 62838-30308)

Address : Chandigarh & Mohali (Punjab), India

Websites: https://www.smartprogramming.in/

https://courses.smartprogramming.in

Android App:
https://play.google.com/store/apps/details?id=com.smartprogramming

YouTube Channel:
https://www.youtube.com/c/SmartProgramming

Smart
Programming
We Educate
We Develop

Smart Programming : YouTube Channel

An investment in Knowledge pays the best interest….

Smart Programming
We Educate - We Develop

☎ +91 62838-30308
Call us to Learn
Latest Technologies

City : Mohali (Punjab),
& Chandigarh
(India)

WEBSITE : http://www.smartprogramming.in

BUY COURSES ON : https://courses.smartprogramming.in

YOUTUBE CHANNEL : Smart Programming (https://www.youtube.com/c/SmartProgramming)

ANDROID APP : Smart Programming
(https://play.google.com/store/apps/details?id=com.smartprogramming)

https://www.facebook.com/smartprogramming.india

https://www.instagram.com/smart_programming

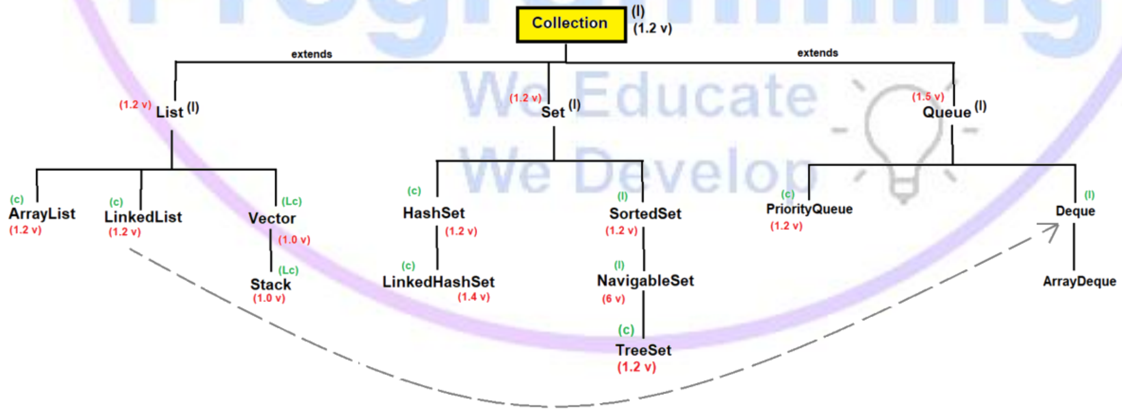
Java
android
python
Data Science
Machine Learning
Artificial Intelligence
M E A N
Full Stack
BIG DATA
hadoop

Queue interface in Java

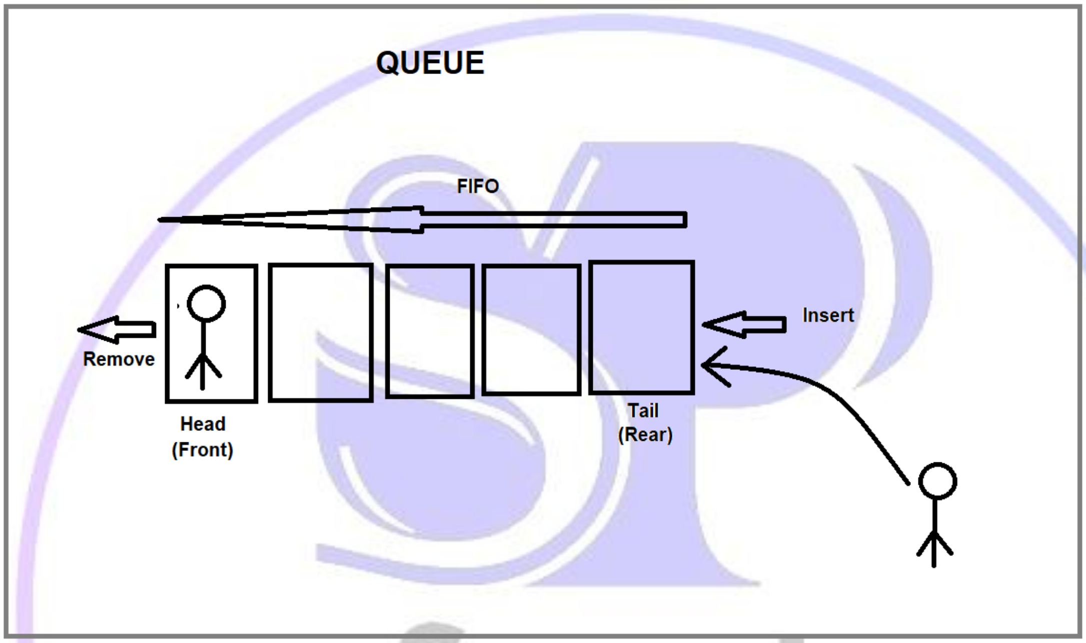
=> Queue :-

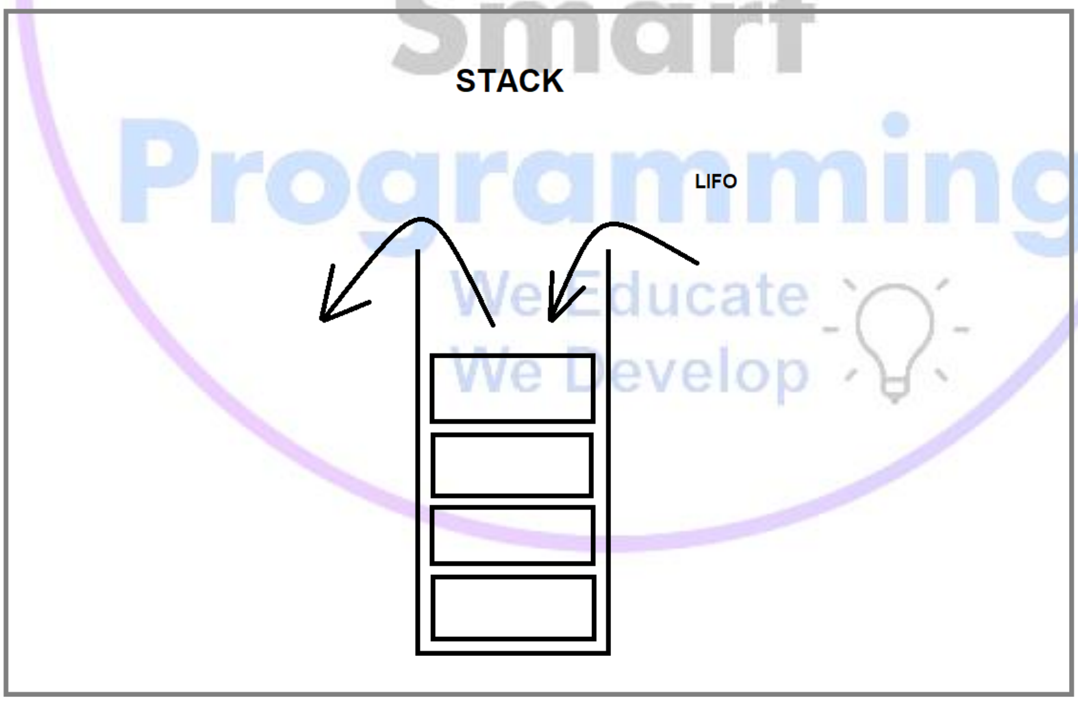
- ➔ Queue is the child interface of Collection interface
- ➔ Syntax : public interface Queue extends Collection { - }
- ➔ Queue was introduced in JDK 1.5 version
- ➔ Queue orders the elements in FIFO(First In First Out) manner, but we can change this algorithm according to our requirements

- ➔ Hierarchy of Queue :-


Collection
(I)
(1.2 v)
extends
extends
(1.2 v) List (I)
(1.2 v) Set (I)
(1.5 v) Queue (I)
(c) ArrayList (1.2 v)
(c) LinkedList (1.2 v)
(Lc) Vector (1.0 v)
(Lc) Stack (1.0 v)
(c) HashSet (1.2 v)
(c) LinkedHashSet (1.4 v)
(I) SortedSet (1.2 v)
(I) NavigableSet (6 v)
(c) TreeSet (1.2 v)
(c) PriorityQueue (1.2 v)
(I) Deque
ArrayDeque

➔ Implementation of Queue :-


QUEUE
FIFO
Remove
Head
(Front)
Tail
(Rear)
Insert



➔ Properties of Queue :-

1. Queue does not follow the insertion order
2. Queue follows the sorting order
3. Queue stores the same data type elements or homogeneous elements. If we try to store different elements then it will throw an exception saying "java.lang.ClassCastException"
4. Queue can store the duplicate elements
5. Queue does not store any null value. If we try to store null value then it will throw an exception i.e. "java.lang.NullPointerException"

➔ Methods of Queue :-

1. boolean offer(Object obj) - to add the elements in the queue
2. Object peek() - It will return the head element of the queue. If no element is found in the queue it will return null value
3. Object element() - It will return the head element of the queue. If no element is found, it

will throw an exception i.e. "java.util.NoSuchElementException"

4. Object poll() - It is used to remove the head element and also it will return that element, If no element is found, then it will return null value
5. Object remove() - It is used to remove the head element and also it will return that element. If no element is found, it will throw an exception i.e. "java.util.NoSuchElementException"

=> PriorityQueue :-

- ➔ PriorityQueue is an implementation class for Queue (but not direct implementation)
- ➔ Syntax : public class PriorityQueue extends AbstractQueue implements Serializable { - }
- ➔ PriorityQueue was introduced in JDK 1.2 version

➔ PriorityQueue may not support on windows platform

➔ It is able to process all the elements prior to processing as per priorities

➔ Properties of PriorityQueue :-

1. PriorityQueue does not follows the insertion order
2. PriorityQueue does not follows the sorting order
3. PriorityQueue stores the same data type elements or homogeneous elements. If we try to store different data type elements then it will throw an exception i.e. "java.lang.ClassCastException"
4. PriorityQueue can stores the duplicate elements
5. PriorityQueue cannot store the null values
6. PriorityQueue is non-synchronized collection

7. PriorityQueue allows more than one thread at one time
8. PriorityQueue allows the parallel execution
9. PriorityQueue reduces the execution time which makes our application fast
10. PriorityQueue is not thread-safe
11. PriorityQueue does not provide guarantee for data consistency

➔ Constructors :-

1. public PriorityQueue() - When we use default PriorityQueue constructor its initial capacity is 11
2. public PriorityQueue(int capacity)
3. public PriorityQueue(Comparator c)
4. public PriorityQueue(int capacity, Comparator c)
5. public PriorityQueue(SortedSet ss)
6. public PriorityQueue(PriorityQueue pq)
7. public PriorityQueue(Collection c)

➔ Methods of PriorityQueue :-

= Contains the methods of Queue and Collection interface

➔ When we should use PriorityQueue :-

= We can use PriorityQueue in SMS (JMS - Java Message Service), mail, offers, prime users etc

=> Deque :

- ➔ It is also known as "double ended queue"
- ➔ In Deque we can add or remove the elements on both side
- ➔ Deque is the child interface of Queue interface
- ➔ Syntax : public interface Deque extends Queue { - }

- ➔ Methods of Deque :-

1. void addFirst(Object e);
2. void addLast(Object e);
3. boolean offerFirst(Object e);
4. boolean offerLast(Object e);
5. Object removeFirst();
6. Object removeLast();
7. Object pollFirst();
8. Object pollLast();
9. Object getFirst();
10. Object getLast();
11. Object peekFirst();
12. Object peekLast();

=> ArrayDeque :-

- ➔ ArrayDeque is an implemented class for Deque interface
- ➔ Syntax : public class ArrayDeque extends AbstractCollection implements Deque, Cloneable, Serializable { - }
- ➔ ArrayDeque is used to provide the facility of Deque and Resizable-Array.

- ➔ Properties of ArrayDeque :-
- ➔ In this also we can add and remove the elements from both side
- ➔ Null is not allowed in ArrayDeque
- ➔ ArrayDeque is not synchronized collection
- ➔ ArrayDeque has no capacity concept

➔ Advantage :-

= ArrayDeque is faster as compared to LinkedList or Stack

Company Links & Contacts

Company Name: Smart Programming (+91 62838-30308)

Address : Chandigarh & Mohali (Punjab), India

Websites: https://www.smartprogramming.in/

https://courses.smartprogramming.in

Android App: https://play.google.com/store/apps/details?id=com.smartprogramming

YouTube Channel: https://www.youtube.com/c/SmartProgramming

Smart Programming : YouTube Channel

An investment in Knowledge pays the best interest….

C
C++
C#
Java
android
python
Data Science
Machine Learning
Artificial Intelligence
M E A N
Full Stack
BIG DATA
hadoop

Smart Programming
We Educate - We Develop

+91 62838-30308
Call us to Learn
Latest Technologies

City : Mohali (Punjab), & Chandigarh (India)

WEBSITE : http://www.smartprogramming.in

BUY COURSES ON : https://courses.smartprogramming.in

YOUTUBE CHANNEL : Smart Programming (https://www.youtube.com/c/SmartProgramming)

ANDROID APP : Smart Programming
(https://play.google.com/store/apps/details?id=com.smartprogramming)

https://www.facebook.com/smartprogramming.india

https://www.instagram.com/smart_programming

Map interface in Java

=> Abstract Design Pattern :

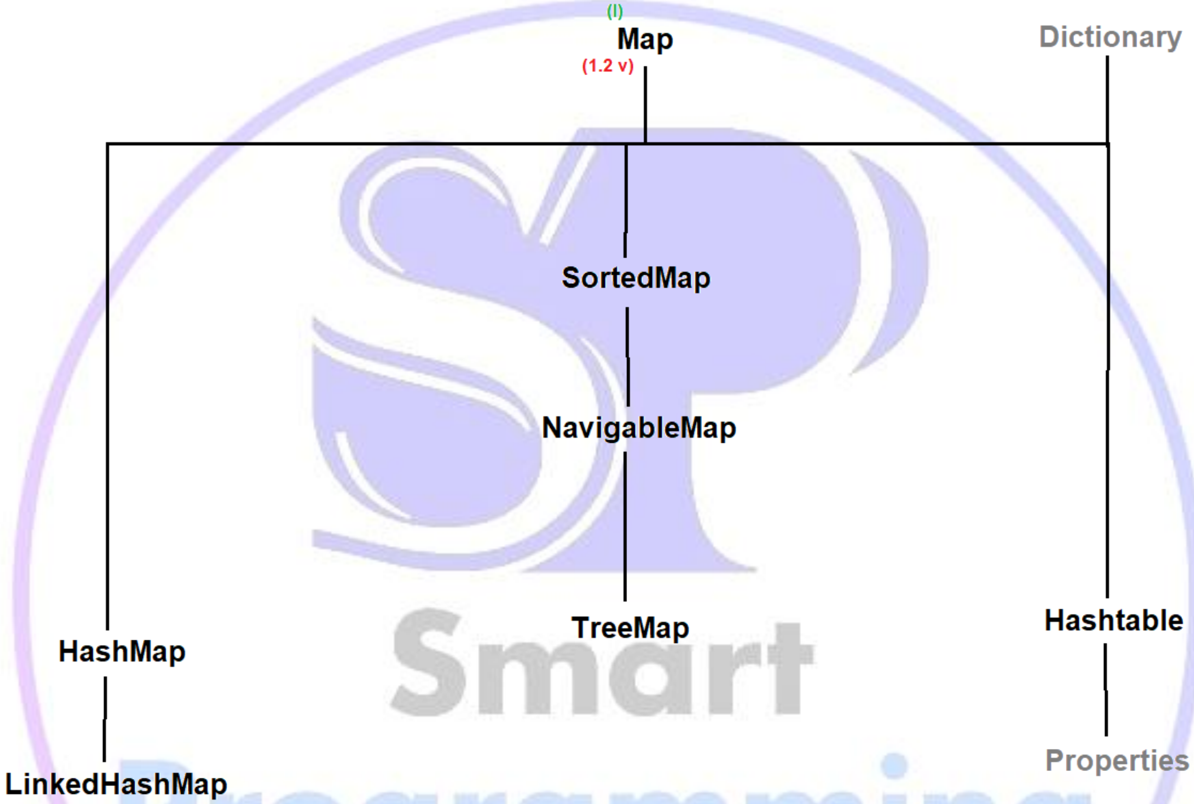
-> It provides a way by which we dont have to implement all the methods of an interface, we just have to declare an abstract class which inherits the interface and in that abstract class we will override the methods of an interface. When we create a class which inherits the abstract class, then we dont need to override all the methods of interface

==

=> Map :-

- Map is an interface which is present in java.util package
- Map does not inherit Collection interface
- Syntax : public interface Map { - }
- Map was introduced in JDK 1.2 version

➔ Hierarchy of Map interface :-


(I)
Map
(1.2 v)
Dictionary
SortedMap
NavigableMap
HashMap
TreeMap
Hashtable
LinkedHashMap
Properties

➔ Properties of Map :-

1. Map stores the data in key-value pair. Each key-value pair is known as Entry
2. In Map, keys should always unique but values can be duplicate
3. Map can store heterogeneous elements or different type of elements

4. In Map keys we can store maximum one null value but in values we can store any number of null values
5. Map does not follows the insertion order by default
6. Map does not follows the sorting order by default

➔ Methods of Map interface :

1. public Object put(Object key, Object value)
2. public void putAll(Map map)
3. public Object get(Object key)
4. public Object remove(Object key)
5. public void clear()
6. public boolean isEmpty()
7. public int size()
8. public boolean containsKey(Object key)
9. public boolean containsValue(Object value)
10. public Set keySet();
11. public Collection values();

Company Links & Contacts

Company Name: Smart Programming (+91 62838-30308)

Address : Chandigarh & Mohali (Punjab), India

Websites: https://www.smartprogramming.in/
https://courses.smartprogramming.in

Android App: https://play.google.com/store/apps/details?id=com.smartprogramming

YouTube Channel: https://www.youtube.com/c/SmartProgramming

Smart Programming : YouTube Channel

An investment in Knowledge pays the best interest….

Smart Programming

We Educate - We Develop

☎ +91 62838-30308

Call us to Learn Latest Technologies

City : Mohali (Punjab), & Chandigarh (India)

WEBSITE : http://www.smartprogramming.in

BUY COURSES ON : https://courses.smartprogramming.in

YOUTUBE CHANNEL : Smart Programming (https://www.youtube.com/c/SmartProgramming)

ANDROID APP : Smart Programming (https://play.google.com/store/apps/details?id=com.smartprogramming)

https://www.facebook.com/smartprogramming.india

https://www.instagram.com/smart_programming

Java
android
python
Data Science
Machine Learning
Artificial Intelligence
M E A N
Full Stack
BIG DATA
hadoop

Entry interface & HashMap class in Java

=> Entry :-

- ➔ Entry is a one key-value pair in Map
- ➔ Without Entry Map is always empty or we can say Map does not contain any key-value pair
- ➔ Entry is an interface which is present in Map interface (or we can say that Entry is a sub-interface of Map interface)
- ➔ Syntax :

```
interface Map
{
        //methods
        interface Entry
        {
                1. Object getKey()
                2. Object getValue()
                3. Object setValue(Object obj)
        }
}
```

=> HashMap :-

- ➔ HashMap is a direct implemented class of Map interface which is present in java.util package
- ➔ Syntax : public class HashMap extends AbstractMap implements Map, Cloneable, Serializable { - }
- ➔ HashMap was introduced in JDK 1.2 version
- ➔ The underline data structure of HashMap is "Hashtable"

➔ Properties of HashMap

1. HashMap stores the values in key-value pair and each key-value pair is known as Entry
2. In HashMap, keys should always unique but values can be duplicate
3. HashMap can store hetrogeneous elements or different type of elements

4. In HashMap keys we can store maximum one null value but in values we can store any number of null values
5. HashMap does not follows the insertion order by default
6. HashMap does not follows the sorting order by default

7. HashMap is non-synchronized Map because HashMap does not contain any synchronized methods
8. HashMap allows more than one thread at one time
9. HashMap allows the parallel execution
10. HashMap reduces the execution time which in turn makes our application fast
11. HashMap is not threadsafe
12. HashMap does not gurantee for data consistency

-> Working of HashMap :-

1. Whenever we create HashMap, its inital capacity is 16 elements
2. HashMap load factor is 75%

➔ Constructors of HashMap :-

1. public HashMap()
2. public HashMap(int capacity)
3. public HashMap(int capacity, float loadFactor)
4. public HashMap(Map m)

➔ Methods of HashMap :-

= same methods as that of Map interface

➔ When we should use HashMap:-

= HashMap is good for searching or retrival operations

➔ How to get synchronized version of HashMap:-

= By default HashMap is non-synchronized but if we want to get synchronized version of HashMap then we have to use synchronizedMap() method of Collections class

Company Links & Contacts

Company Name: Smart Programming (+91 62838-30308)

Address : Chandigarh & Mohali (Punjab), India

Websites: https://www.smartprogramming.in/

https://courses.smartprogramming.in

Android App:
https://play.google.com/store/apps/details?id=com.smartprogramming

YouTube Channel:
https://www.youtube.com/c/SmartProgramming

Smart Programming : YouTube Channel

An investment in Knowledge pays the best interest....

Smart Programming
We Educate - We Develop

☎ +91 62838-30308
Call us to Learn Latest Technologies

City : Mohali (Punjab), & Chandigarh (India)

WEBSITE : http://www.smartprogramming.in

BUY COURSES ON : https://courses.smartprogramming.in

YOUTUBE CHANNEL : Smart Programming (https://www.youtube.com/c/SmartProgramming)

ANDROID APP : Smart Programming (https://play.google.com/store/apps/details?id=com.smartprogramming)

https://www.facebook.com/smartprogramming.india

https://www.instagram.com/smart_programming

Java
android
python
Data Science
Machine Learning
Artificial Intelligence
M E A N
Full Stack
BIG DATA
hadoop

LinkedHashMap, IdentityHashMap & WeakHashMap class in Java

=> LinkedHashMap :-

- ➔ LinkedHashMap is the child class of HashMap which is present in java.util package
- ➔ Syntax : public class LinkedHashMap extends HashMap implements Map { - }
- ➔ LinkedHashMap was introduced in JDK 1.4 version
- ➔ LinkedHashMap underline data structure is "Hashtable + LinkedList"

➔ Properties of LinkedHashMap :-

= LinkedHashMap has the same properties as that of HashMap but one difference is LinkedHashMap follows the insertion order

➔ Constructors of LinkedHashMap :-

= Same constructors as that of HashMap

➔ Methods of LinkedHashMap :-

= Same methods as that of HashMap

➔ When we should use LinkedHashMap :-

= If we have to create cache based applications then we can use LinkedHashMap

=> What is difference between HashMap & LinkedHashMap :-

1. HashMap was introduced in JDK 1.2 version
   LinkedHashMap was introduced in JDK 1.4 version

2. HashMap does not follows the insertion order
   LinkedHashMap follows the insertion order

3. HashMap uderline data structure is "Hashtable"
   LinkedHashMap underline data structure is "Hashtable + LinkedList"

=> Difference between == & .equals() method :-

== is used for reference comparision or address comparision

equals() method is used for content comparision

=> IdentityHashMap :-

- ➔ IdentityHashMap is an implemented class of Map interface which is present in java.util package
- ➔ Syntax : public class IdentityHashMap extends AbstractMap implements Map, Serializable, Cloneable { - }
- ➔ IdentityHashMap was introduced in JDK 1.4 version

➔ **Properties, Constructors & Methods are same as HashMap**

➔ **Point to remember :-**

1. Map does not contain duplicate keys but can contain duplicate values
2. If we insert key-value pair in HashMap internally .equals method will be called and keys will be compared, if the value is true, it will treat as duplicate key and it will not insert that key

3. If we insert key-value pair in IdentityHashMap, then internally == operator will be called, and keys will be compared, if the value is true, then it will replace the value or if the value is false then it will insert that key-value pair

⇨ Difference between HashMap & IdentityHashMap :-

1. HashMap was introduced in JDK 1.2 version
   IdentityHashMap was introduced in JDK 1.4 version

2. In case of HashMap to check for duplicate keys, internally .equals () method is called
   In case of IdentityHashMap, to check for duplicate keys, internally == operator will be called

---

=> How to delete an object in java ?

- ➔ There are 2 steps to delete an object in java :-
  1. Provide null value to the reference
  2. Call gc() method

=> WeakHashMap :-

- ➔ WeakHashMap is a class which implements the Map interface
- ➔ Syntax : Public class WeakHashMap extends AbstractMap implements Map { - }
- ➔ WeakHashMap was introduced in JDK 1.2 version

- ➔ **Properties, Constructors and methods are same as HashMap**

=> Difference between HashMap & WeakHashMap :-

1. HashMap does not allow the garbage collector to delete its elements

WeakHashMap allows the garbage collector to delete its elements

Company Links & Contacts

Company Name: Smart Programming (+91 62838-30308)

Address : Chandigarh & Mohali (Punjab), India

Websites: https://www.smartprogramming.in/

https://courses.smartprogramming.in

Android App: https://play.google.com/store/apps/details?id=com.smartprogramming

YouTube Channel: https://www.youtube.com/c/SmartProgramming

Smart Programming : YouTube Channel

An investment in Knowledge pays the best interest….

Smart Programming
We Educate - We Develop

☎ +91 62838-30308
Call us to Learn
Latest Technologies

City : Mohali (Punjab),
& Chandigarh
(India)

WEBSITE : http://www.smartprogramming.in

BUY COURSES ON : https://courses.smartprogramming.in

YOUTUBE CHANNEL : Smart Programming (https://www.youtube.com/c/SmartProgramming)

ANDROID APP : Smart Programming
(https://play.google.com/store/apps/details?id=com.smartprogramming)

https://www.facebook.com/smartprogramming.india

https://www.instagram.com/smart_programming

Java
android
python
Data Science
Machine Learning
Artificial Intelligence
M E A N
Full Stack
BIG DATA
hadoop

SortedMap, NavigableMap & TreeMap in Java

=> SortedMap :-

- ➔ SortedMap is a child interface of Map interface which is present in java.util package
- ➔ Syntax : public interface SortedMap extends Map { - }
- ➔ SortedMap was introduced in JDK 1.2 version

➔ Properties of SortedMap :-

1. SortedMap stores the data in key-value pair where key must be unique but values can be duplicate
2. SortedMap does not follows the insertion order w.r.t. keys
3. SortedMap follows the sorting order w.r.t. keys
4. SortedMap can store homogeneous and heterogeneous keys :-
    a. If we are depending on default natural sorting order then the keys should be homogeneous and Comparable otherwise it will provide an exception saying "java.lang.ClassCastException"

   b. If we are defigning our own sorting order by Comparator, then the keys can be heterogeneous.

5. We cannot store null values in SortedMap

6. SortedMap is non-synchronized Map

➔ Methods of SortedMap :-

1. Object firstKey();
2. Object lastKey();
3. SortedMap headMap(Object key);
4. SortedMap tailMap(Object key);
5. SortedMap subMap(Object key2, Object key2);
6. Comparator comparator();

=> NavigableMap :-

- ➔ NavigableMap is the child interface is SortedMap which is present in java.util package
- ➔ Syntax : public interface NavigableMap extends SortedMap { - }
- ➔ NavigableMap was introduced in Java SE 6 version

➔ Properties of NavigableMap :-

= NavigableMap is same as SortedMap but it provides some extra navigable methods

➔ Methods of NavigableMap :-

1. public NavigableMap descendingMap()
2. public Object ceilingKey(Object key)
3. public Object higherKey(Object key)
4. public Object floorKey(Object key)
5. public Object lowerKey(Object key)
6. public void pollFirstEntry()
7. public void pollLastEntry()

=> TreeMap :-

- ➔ TreeMap is a clas which provides the implementation for NavigableMap, SortedMap & Map interface
- ➔ Syntax : public class TreeMap extends AbstractMap implements NavigableMap, Cloneable, Serializable { - }
- ➔ TreeMap was introduced in JDK 1.2 version
- ➔ The underline data structure of TreeMap is "Red-Black tree"

- ➔ Properties of TreeMap :-

1. TreeMap stores the data in key-value pair which key must be unique but value can be duplicate
2. TreeMap does not follows the insertion order wrt keys
3. TreeMap follows the sorting order wrt keys

4. We can store homogeneous & heterogeneous elements :-
    a. If we are depending on default natural sorting order then the keys must be homogeneous and comparable
    b. If we are defining our own sorting by Comparator then the keys can be heterogeneous
5. We cannot store the null values in TreeMap

6. TreeMap is non-synchronized Map
7. TreeMap allows more than one thread at one time
8. TreeMap allows the parallel execution
9. TreeMap reduces the execution time which makes our application fast
10. TreeMap are not thread-safe
11. TreeMap does not provide guarantee for data consistency

➔ Constructors :-

1. public TreeMap() { - }
2. public TreeMap(Comparator comparator) { - }
3. public TreeMap(Map m) { - }
4. public TreeMap(SortedMap m) { - }

➔ Methods :-

= implements all the methods of Map, SortedMap & NavigableMap interface

➔ When we should use TreeMap :-

= When we want to store a large number of elements in sorting order then we have to use TreeMap which makes the retrival operation fast

➔ "null" value insertion in TreeMap :-

1. Till 1.6 version we were able to insert null value at first key position but if we insert any other value then it will provide "NullPointerException"
2. If we insert null values after first position then it will throw an exception saying "NullPointerException"
3. From 1.7 version we cannot insert the null value even at first position

---

=> What is difference between HashMap, LinkedHashMap & TreeMap :-

1. HashMap was introduced in JDK 1.2 version
   LinkedHashMap was introduced in JDK 1.4 version
   TreeMap was introduced in JDK 1.2 version

2. HashMap underline data-structure is "Hashtable"
   LinkedHashMap underline data-structure is "Hashtable+LinkedList"
   TreeMap underline data-structure is "Red-Black Tree"

3. HashMap does not follows the insertion order
   LinkedHashMap follows the insertion order
   TreeMap does not follows the insertion order

4. HashMap does not follows the sorting order
   LinkedHashMap does not follows the sorting order
   TreeMap follows the sorting order

5. HashMap allows the heterogeneous objects by default
   LinkedHashMap allows heterogeneous objects by default

TreeMap allows homogeneous objects by default

6. HashMap allows the null value insertion
   LinkedHashMap allows the null value insertion
   TreeMap does not allows the null value insertion

Company Links & Contacts

Company Name: Smart Programming (+91 62838-30308)

Address : Chandigarh & Mohali (Punjab), India

Websites: https://www.smartprogramming.in/
https://courses.smartprogramming.in

Android App: https://play.google.com/store/apps/details?id=com.smartprogramming

YouTube Channel: https://www.youtube.com/c/SmartProgramming

Smart Programming : YouTube Channel

An investment in Knowledge pays the best interest….

Smart Programming
We Educate - We Develop

☎ +91 62838-30308
Call us to Learn Latest Technologies

City : Mohali (Punjab), & Chandigarh (India)

WEBSITE : http://www.smartprogramming.in

BUY COURSES ON : https://courses.smartprogramming.in

YOUTUBE CHANNEL : Smart Programming (https://www.youtube.com/c/SmartProgramming)

ANDROID APP : Smart Programming (https://play.google.com/store/apps/details?id=com.smartprogramming)

https://www.facebook.com/smartprogramming.india

https://www.instagram.com/smart_programming

Java
android
python
Data Science
Machine Learning
Artificial Intelligence
M E A N
Full Stack
BIG DATA
hadoop

Hashtable & Working of Hashtable & HashMap in Java

=> Hashtable :-

- ➔ Hashtable is a direct implemented class of Map interface which is present in java.util package
- ➔ Syntax : public class Hashtable extends Dictionary implements Map, Cloneable, Serializable { - }
- ➔ Hashtable was introduced in JDK 1.0 version
- ➔ Hashtable is also known as legacy class
- ➔ The underline data structure of Hashtable is "Hashtable"

➔ Properties of Hashtable :-

1. Hashtable stores the values in key-value pair and each key-value pair is known as entry
2. In Hashtable, keys should always unique but values can be duplicate
3. Hashtable can store heterogeneous elements at key position
4. In Hashtable we cannot insert null values at key or value position

5. Hashtable does not follows the insertion order by default
6. Hashtable does not follows the sorting order by default

7. Hashtable is synchronized map
8. Hashtable does not allows more than one thread at one time
9. Hashtable allows the sequential execution
10. Hashtable increases the execution time which in turn makes our application slow
11. Hashtable is thread-safe
12. Hashtable provides the guarantee for data consistency

➔ Constructors :-

1. 1. public Hashtable() { - }   //it will create a Hashtable having 11 as capacityand default fill ratio is .75%
2. public Hashtable(Map t) { - }
3. public Hashtable(int initialCapacity) { - }

4. public Hashtable(int initialCapacity, float loadFactor) { - }

- ➔ Methods :-

  = same methods as that of Map interface

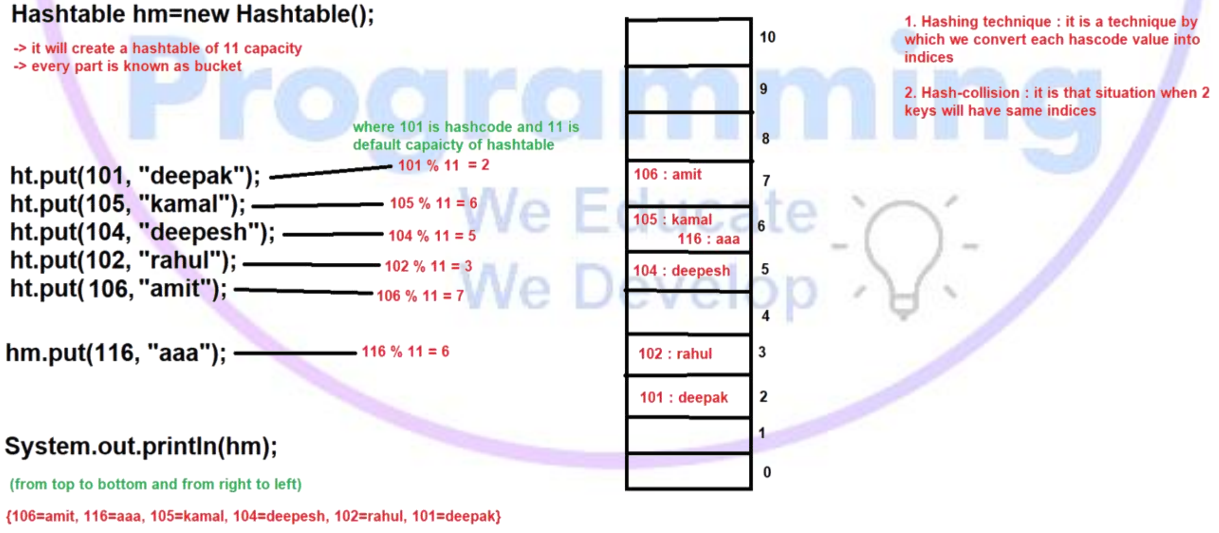
- ➔ When we should use Hashtable :-

  = Hashtable is good for searching or retrieval operation

=> Working of Hashtable :-

- ➔ "hashcode" is the unique integer value of each and every object that is provided by JVM
- ➔ Hashtable initialCapacity is 11

- ➔ Then for each and every key hashcode value will be generated and its index position will be calculated by using hashing technique
- ➔ And at that index position that key-value pair or entry will be inserted
- ➔ If two elements have same index position, then that entry will be inserted at right side of previous entry
- ➔ When the values are traversed then they are traversed from top to bottom and right to left



```
Hashtable hm=new Hashtable();
```

-> it will create a hashtable of 11 capacity
-> every part is known as bucket

where 101 is hashcode and 11 is default capaicty of hashtable

```
ht.put(101, "deepak");
ht.put(105, "kamal");
ht.put(104, "deepesh");
ht.put(102, "rahul");
ht.put( 106, "amit");

hm.put(116, "aaa");

System.out.println(hm);
```

(from top to bottom and from right to left)

```
{106=amit, 116=aaa, 105=kamal, 104=deepesh, 102=rahul, 101=deepak}
```

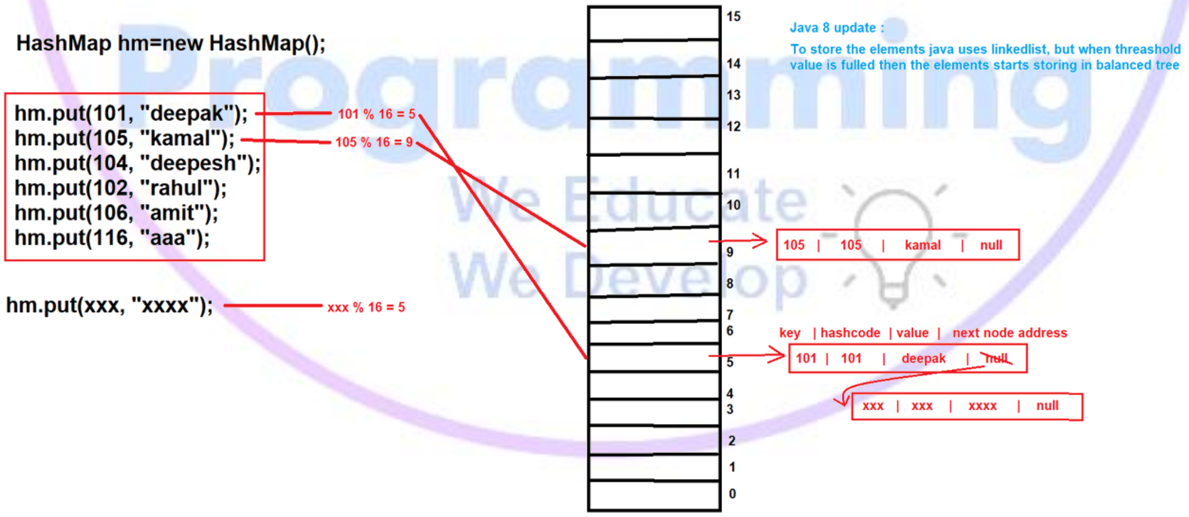
1. Hashing technique : it is a technique by which we convert each hascode value into indices

2. Hash-collision : it is that situation when 2 keys will have same indices

101 % 11 = 2
105 % 11 = 6
104 % 11 = 5
102 % 11 = 3
106 % 11 = 7
116 % 11 = 6
10
9
8
106 : amit
7
105 : kamal
116 : aaa
6
104 : deepesh
5
4
102 : rahul
3
101 : deepak
2
1
0

=> Working of HashMap :-

- ➔ HashMap initialCapacity is 16
- ➔ For every entry it will calculate the index position and store the element there.
- ➔ If multiple entries have same index position, then it will create linked list and starts storing in that linked-list
- ➔ In java 8 updates, after threshold value is filled then it starts storing the elements in balanced tree



```
HashMap hm=new HashMap();

hm.put(101, "deepak");
hm.put(105, "kamal");
hm.put(104, "deepesh");
hm.put(102, "rahul");
hm.put(106, "amit");
hm.put(116, "aaa");

hm.put(xxx, "xxxx");
```

Java 8 update :

To store the elements java uses linkedlist, but when threashold value is fulled then the elements starts storing in balanced tree

101 % 16 = 5
105 % 16 = 9
xxx % 16 = 5
15
14
13
12
11
10
9
8
7
6
5
4
3
2
1
0
105 | 105 | kamal | null
key | hashcode | value | next node address
101 | 101 | deepak | null
xxx | xxx | xxxx | null

=> Difference between HashMap & Hashtable :-

1. HashMap was introduced in 1.2 version
   Hashtable was introduced in 1.0 version

2. HashMap is not a legacy class
   Hashtable is a legacy class

3. In HashMap we can store the null values
   In Hashtable we cannot store the null value at key or value position

4. HashMap is non-synchronized Map because HashMap does not contain any synchronized methods
   Hashtable is synchronized Map because it contains synchronized methods

5. Multiple points related to synchronization

Company Links & Contacts

Company Name: Smart Programming (+91 62838-30308)

Address : Chandigarh & Mohali (Punjab), India

Websites: https://www.smartprogramming.in/

https://courses.smartprogramming.in

Android App:
https://play.google.com/store/apps/details?id=com.smartprogramming

YouTube Channel:
https://www.youtube.com/c/SmartProgramming

Smart Programming : YouTube Channel

An investment in Knowledge pays the best interest….

Smart Programming
We Educate - We Develop

+91 62838-30308
Call us to Learn
Latest Technologies

City : Mohali (Punjab),
& Chandigarh
(India)

WEBSITE : http://www.smartprogramming.in

BUY COURSES ON : https://courses.smartprogramming.in

YOUTUBE CHANNEL : Smart Programming (https://www.youtube.com/c/SmartProgramming)

ANDROID APP : Smart Programming
(https://play.google.com/store/apps/details?id=com.smartprogramming)

https://www.facebook.com/smartprogramming.india

https://www.instagram.com/smart_programming

Java
android
python
Data Science
Machine Learning
Artificial Intelligence
M E A N
Full Stack
BIG DATA
hadoop

Dictionary, Properties & Concurrent Classes in Java

=> Dictionary :-

- ➔ Dictionary is an abstract class which is present in java.util package
- ➔ Dictionary represents the key-value pair similar to Map interface
- ➔ Syntax : public abstract class Dictionary { - }
- ➔ As Dictionary is an abstract class, thus we cannot create an object of it

- ➔ Methods :-

1. abstract public int size();
2. abstract public boolean isEmpty();
3. abstract public Enumeration keys();
4. abstract public Enumeration elements();
5. abstract public Object get(Object key);
6. abstract public Object put(Object key, Object value);
7. abstract public Object remove(Object key);

- ➔ Dictionary was introduced in JDK 1.0 version
- ➔ Since java does not support multiple inheritance, if any class extends the Dictionary class, then it will not be able to inherit any other class and due to this reason, Map interface was created.
- ➔ Dictionary class is not used that much as Map interface.

=> Properties :-

- ➔ Properties is the child class of Hashtable which is present in java.util package
- ➔ Syntax : public class Properties extends Hashtable { - }
- ➔ Properties is the file which stores the data in key-value pair
- ➔ Properties can contain "key-value pair in the form of String type only"

➔ When we should use Properties :-

- Whenever there is any data which can change frequently in future then we should not embed that data in our java application. Instead of that we have to create properties file and provide the data in key-value pair so that we dont need to recompile, rebuild, redeploye and even no need to restart the server
- For example validations, database connection, exception handling etc

➔ Constructors :-

1. public Properties() { - }
2. public Properties(Properties defaults) { - }

➔ Methods :-

1. void load(InputStream is)
2. String getProperty(String pname)
3. String setProperty(String pname, pvalue);
4. void store(OutputStream os, String comments)

=> Concurrent Collections :-

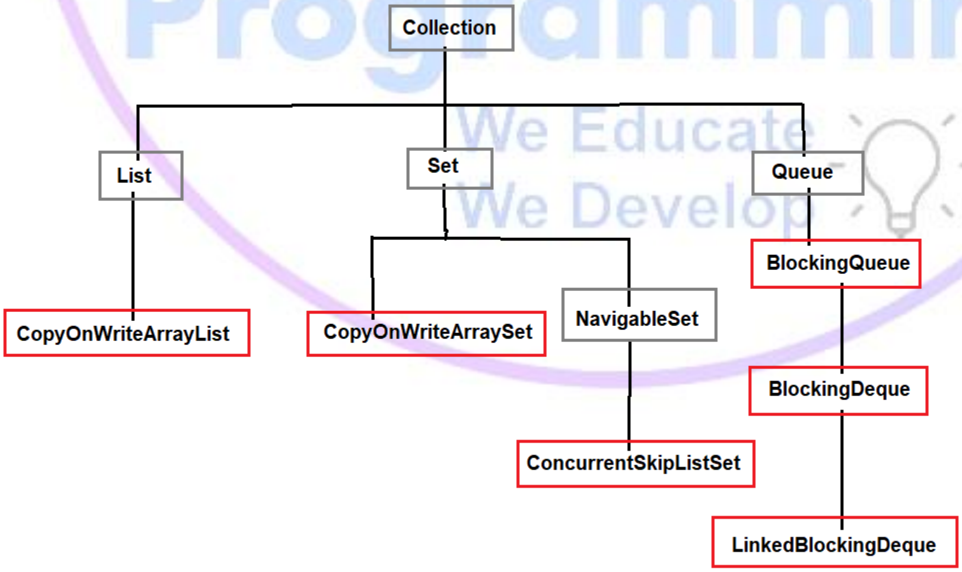
➔ Why Concurrent Collections were introduced :-

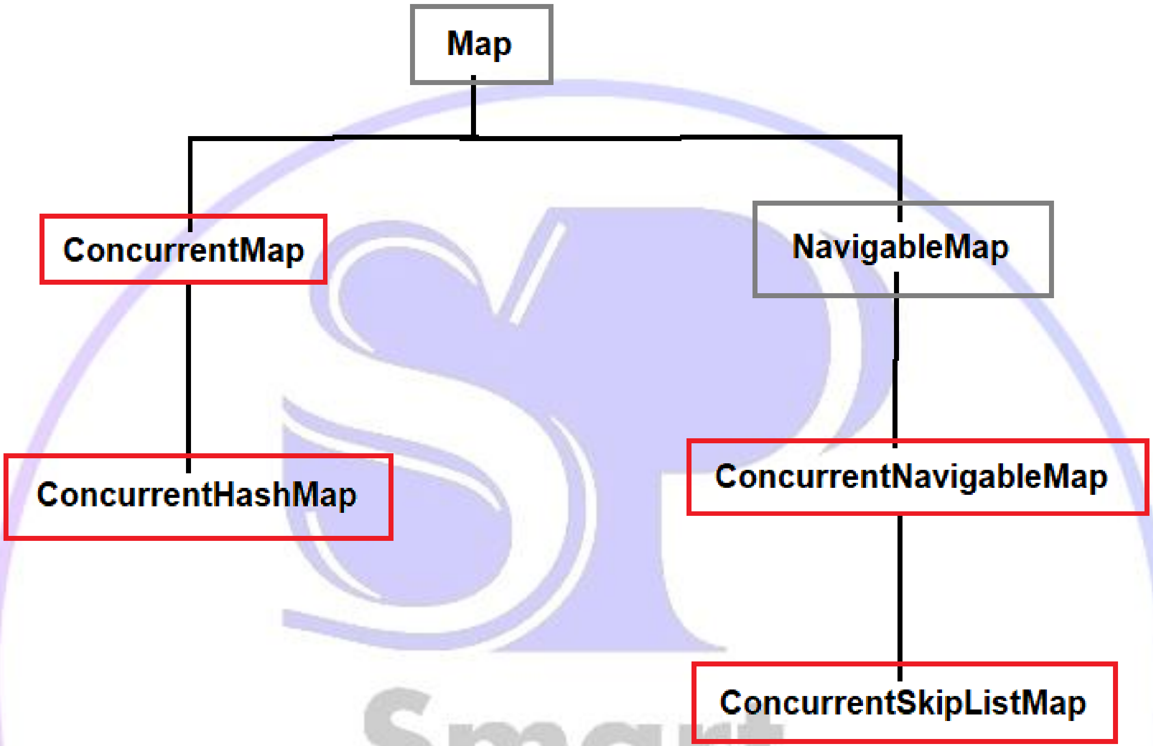
- Most of the classes that we have used till now (traditional collections) are non-synchronized for example ArrayList, LinkedList, HashSet, HashMap etc
- If any class is non-synchronized then it does not guarantee for data consistency which can ruin our application
- Some of the synchronized collection like Vector or Stack have low performace
- In traditional collections, if a thread is iterating a Collection object and if another thread try to add new element in that iterating object, then it will provide an exception saying "ConcurrentModificationException"

- ➔ Due to above problems, traditional Collections were not good for multithreaded applications and due to this, concurrent collections were introduced

- ➔ The concurrent Collections were introduced in JDK 1.5 version
- ➔ These Concurrent Collection classes are present in java.util.concurrent package

- ➔ Hierarchy of Concurrent Collection :-


Collection
List
Set
Queue
CopyOnWriteArrayList
CopyOnWriteArraySet
NavigableSet
ConcurrentSkipListSet
BlockingQueue
BlockingDeque
LinkedBlockingDeque


Map
ConcurrentMap
NavigableMap
ConcurrentHashMap
ConcurrentNavigableMap
ConcurrentSkipListMap

Company Links & Contacts

Company Name: Smart Programming (+91 62838-30308)

Address : Chandigarh & Mohali (Punjab), India

Websites: https://www.smartprogramming.in/

https://courses.smartprogramming.in

Android App: https://play.google.com/store/apps/details?id=com.smartprogramming

YouTube Channel: https://www.youtube.com/c/SmartProgramming

Smart Programming : YouTube Channel

An investment in Knowledge pays the best interest….

Smart Programming
We Educate - We Develop

☎ +91 62838-30308
Call us to Learn
Latest Technologies
City : Mohali (Punjab),
& Chandigarh
(India)

WEBSITE : http://www.smartprogramming.in

BUY COURSES ON : https://courses.smartprogramming.in

YOUTUBE CHANNEL : Smart Programming (https://www.youtube.com/c/SmartProgramming)

ANDROID APP : Smart Programming
(https://play.google.com/store/apps/details?id=com.smartprogramming)

https://www.facebook.com/smartprogramming.india

https://www.instagram.com/smart_programming

Java
android
python
Data Science
Machine Learning
Artificial Intelligence
M E A N
Full Stack
BIG DATA
hadoop

Generics in Java

=> Generics :-

- ➔ Generics means parametrized types which means that we can provide any type of parameter to the classes, interfaces or methods
- ➔ Generics were introduced in JDK 1.5 version
- ➔ Generics are represented by angular braces - < >
- ➔ The main objective of Generics are :-

1. To provide type safety
2. To resolve type casting problem

- ➔ By default "arrays are type safe"
- ➔ Now for collections, till JDK 1.4 version, collections were not generic types
- ➔ In JDK 1.5 version, Generic Collections were introduced

- ➔ NOTE : We can only provide non-primitive values in generics

---

=> Generic Classes :-

- ➔ If any class is declared with type parameters then it is known as Generic Class
- ➔ Generic classes can be user-defined classes or predefined classes (collections classes)

- ➔ Generic type can be any valid identifier name
- ➔ We can provide any number of parameters in generics

=> Generic Bounded Types :-

- ➔ We can bound the type parameter for a particular range by using extends keyword. And this concept is known as Generic Bounded Type Concept

➔ Syntax : class A<T extends X>
(X can be any class or interface)

➔ Cases :-

1. We can only use extends keyword, not implements keyword
2. We can only use Non-Primitive data types
3. class A<T extends X & Y>

Company Links & Contacts

Company Name: Smart Programming (+91 62838-30308)

Address : Chandigarh & Mohali (Punjab), India

Websites: https://www.smartprogramming.in/

https://courses.smartprogramming.in

Android App: https://play.google.com/store/apps/details?id=com.smartprogramming

YouTube Channel: https://www.youtube.com/c/SmartProgramming

Smart Programming : YouTube Channel

An investment in Knowledge pays the best interest....

Smart Programming
We Educate - We Develop

☎ +91 62838-30308
Call us to Learn
Latest Technologies

City : Mohali (Punjab),
& Chandigarh
(India)

WEBSITE : http://www.smartprogramming.in

BUY COURSES ON : https://courses.smartprogramming.in

YOUTUBE CHANNEL : Smart Programming (https://www.youtube.com/c/SmartProgramming)

ANDROID APP : Smart Programming
(https://play.google.com/store/apps/details?id=com.smartprogramming)

https://www.facebook.com/smartprogramming.india

https://www.instagram.com/smart_programming

Java
android
python
Data Science
Machine Learning
Artificial Intelligence
M E A N
Full Stack
BIG DATA
hadoop

Java 8 Features

=> Java Version Features :-

➔ JDK 1.0 & 1.1 Version :-

- Java was created
- AWT event model
- Inner classes
- JavaBeans
- JDBC
- Reflection
- RMI
- JIT compiler for windows platform

➔ JDK 1.2 Version :-

- Collection frameworks
- strictfp keyword
- Swing API
- Java Plugins

➔ JDK 1.3 Version :-

- HotStop JVM
- JNDI (Java Naming & DIrectory Interface)

  - JavaSound

➔ JDK 1.4 Version :-

  - RegularExpression (RegEx)
  - Exception Chaining
  - NIO
  - Logging API
  - Image IO API

➔ JDK 1.5 Version :-

  - Generics
  - Annotations
  - Autoboxing/Unboxing
  - Enumerations
  - Varagrs
  - Enhanced for-each loop
  - New concurrent utilities in java.util.concurrent package

➔ Java SE 6 Version :-

- JDBC 4.0
- Java Compiler API
- Support for Scripting Langauges
- Performance updates
- New GC algorithms

➔ Java SE 7 Version :-

- Strings in switch
- JVM supports the dynamic languages
- Compressed 64-bit pointers

➔ Java SE 8 Version :-

- Default Methods in Interface
- Static Methods in Interface
- Functional Interfaces
- Lambda Expressions
- Pre-defined Functional Interfaces (Predicate, Function, Consumer, Supplier etc)
- Stream API

- ○ Method Reference
- ○ Constructor Reference
- ○ Date-Time API

=> Why Java SE 8 version is important :-

1. It simplifies the java programming

2. Lambda expressions inherits the features/benifits of "functional programming languages" (But still java is OOP language)

3. Java supports parallel programming

4. Today we have only multiple core processors and java also starts utilizing mutiple core functionalities

---

=> Default Methods in Interface :-

- ➔ Interface can contain only abstract methods which does not have any implementation part and its implementations where provided in the implemented class
- ➔ But from 1.8 version, we can create default methods having implementation part in interface. And if we are not satisfied with default method implementation, then we can override that method and provide its our new implementation.
- ➔ Default methods are created by using "default" keyword
- ➔ Note :

1. We can override default methods
2. We can provide public access-modifier to default methods

➔ Use of default methods :-

1. If we change the interface then it will affect all the implemented classes, but by creating default methods, implemented classes will not be affected.
2. By creating default methods we can provide same implementation for multiple classes in the interface which in turn reduces the number of lines of code

---

=> Static Methods in Interface :-

- ➔ Till 1.7 version we were not able to create any static method in an interface but from 1.8 version we can create static methods in an interface
- ➔ static methods have implementation part in the interface only

➔ Static methods are created by using "static" keyword

➔ **Note :**

1. We can use public access-modifier with static method
2. We cannot use default keyword with static method
3. We were not able to override static methods. If we try to override it, it will be treated as different method

➔ Static methods are used to improve the shareability in interface

---

=> Marker Interface :-

- ➔ If any interface does not contain any abstract method, then it is known as marker interface
- ➔ Predefined marker interfaces are Cloneable, Serializable, Remote etc

=> Functional Interfaces :-

- ➔ If any interface contains only and only one abstract method, then it is known as Functional Interface
- ➔ To denote function interface we use "@FunctionalInterface" annotation
- ➔ Cases for Functional Interface :-

1. We can create any number of "default" or "static" methods in functional interface but we can create only and only one asbtract method
2. In case of inheritance if any interface have more than one abstract method, then also it will not be treated as Functional Interface

- ➔ Predefined functional interfaces are :- Comparable, ActionListener, Runnable, Callable etc

=> Annotations :-

- ➔ Annotations are used to provide some additional information about the program
- ➔ Annotations starts with "@"
- ➔ If we use or dont use annotations then there is no effect on the compiler
- ➔ Annotations provides some metadata for classes or interfaces or methods or variables or constructors

Company Links & Contacts

Company Name: Smart Programming (+91 62838-30308)

Address : Chandigarh & Mohali (Punjab), India

Websites: https://www.smartprogramming.in/

https://courses.smartprogramming.in

Android App:
https://play.google.com/store/apps/details?id=com.smartprogramming

YouTube Channel:
https://www.youtube.com/c/SmartProgramming

Smart Programming : YouTube Channel

An investment in Knowledge pays the best interest....

Smart Programming
We Educate - We Develop

☎ +91 62838-30308
Call us to Learn
Latest Technologies

City : Mohali (Punjab), & Chandigarh (India)

WEBSITE : http://www.smartprogramming.in

BUY COURSES ON : https://courses.smartprogramming.in

YOUTUBE CHANNEL : Smart Programming (https://www.youtube.com/c/SmartProgramming)

ANDROID APP : Smart Programming (https://play.google.com/store/apps/details?id=com.smartprogramming)

https://www.facebook.com/smartprogramming.india

https://www.instagram.com/smart_programming

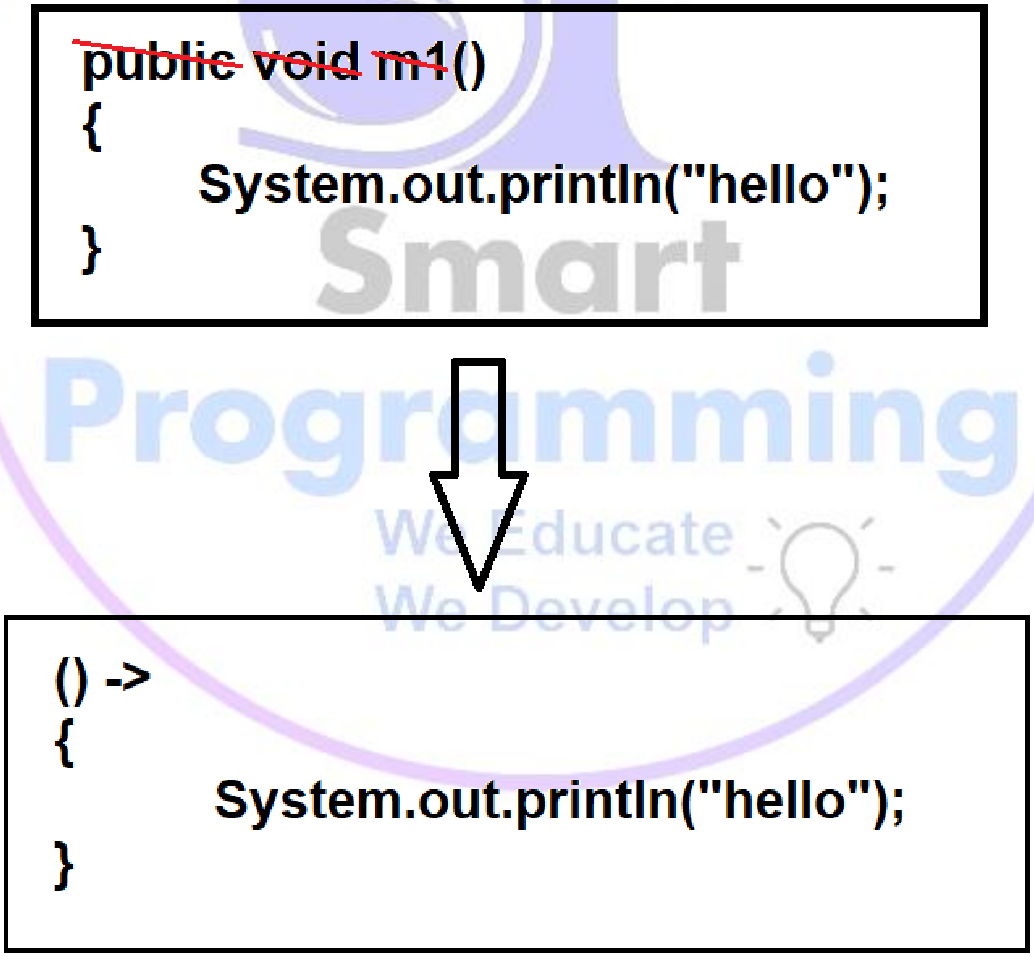
Java
android
python
Data Science
Machine Learning
Artificial Intelligence
M E A N
Full Stack
BIG DATA
hadoop

Lambda Expression in Java

=> Lambda Expressions :-

➔ Lambda is a function which does not have

- any name
- any modifier
- return type



```
~~public void m1~~()
{
        System.out.println("hello");
}
```

```
() ->
{
        System.out.println("hello");
}
```

➔ Advantages of Lambda Expressions :-

1. It reduces the number of lines of code
2. It can call the API very effectively
3. It supports parallel and sequentical execution
4. It is used to write readable, maintanable and concise code

➔ NOTE : "Lambda Expressions" are always used with "Functional Interfaces"

➔ Rules related to lambda functions :-

1. If we have only one statement in lambda expression then we can remove curly braces - {}
2. Lambda expression can take any number of parameters
3. Providing data type with parameters in lambda expression is optional. If we dont provide then compiler automatically deducts the type of data based on context.
4. If we have only one parameter then we can remove round braces - ()

5. If lambda expression is returning any value, then we can remove "return" keyword
6. We can provide any number of statements in lambda expression

---

Company Links & Contacts

Company Name: Smart Programming (+91 62838-30308)

Address : Chandigarh & Mohali (Punjab), India

Websites: https://www.smartprogramming.in/

https://courses.smartprogramming.in

Android App:
https://play.google.com/store/apps/details?id=com.smartprogramming

YouTube Channel:
https://www.youtube.com/c/SmartProgramming

Smart Programming : YouTube Channel

An investment in Knowledge pays the best interest….

Smart Programming

We Educate - We Develop

☎ +91 62838-30308

Call us to Learn Latest Technologies

City : Mohali (Punjab), & Chandigarh (India)

WEBSITE : http://www.smartprogramming.in

BUY COURSES ON : https://courses.smartprogramming.in

YOUTUBE CHANNEL : Smart Programming (https://www.youtube.com/c/SmartProgramming)

ANDROID APP : Smart Programming (https://play.google.com/store/apps/details?id=com.smartprogramming)

https://www.facebook.com/smartprogramming.india

https://www.instagram.com/smart_programming

Java
android
python
Data Science
Machine Learning
Artificial Intelligence
M E A N
Full Stack
BIG DATA
hadoop

Predefined Functional Interfaces

=> Predicate :-

- ➔ Predicate is a functional interface which is present in java.util.function package
- ➔ Syntax :-

```
public interface Predicate<T>
{
        boolean test(T t);
        //some default methods are also present
}
```

- ➔ It improves manageability of the code and also helps in unit-testing

=> Function :-

- ➔ Function is a functional interface which is present in java.util.function package
- ➔ Syntax :

```
public interface Function<T,R>
{
    R apply(T t);
    //some default methods are also present
}
```

=> Consumer :-

- ➔ Consumer is a functional interface which is present in java.util.function package
- ➔ Syntax :

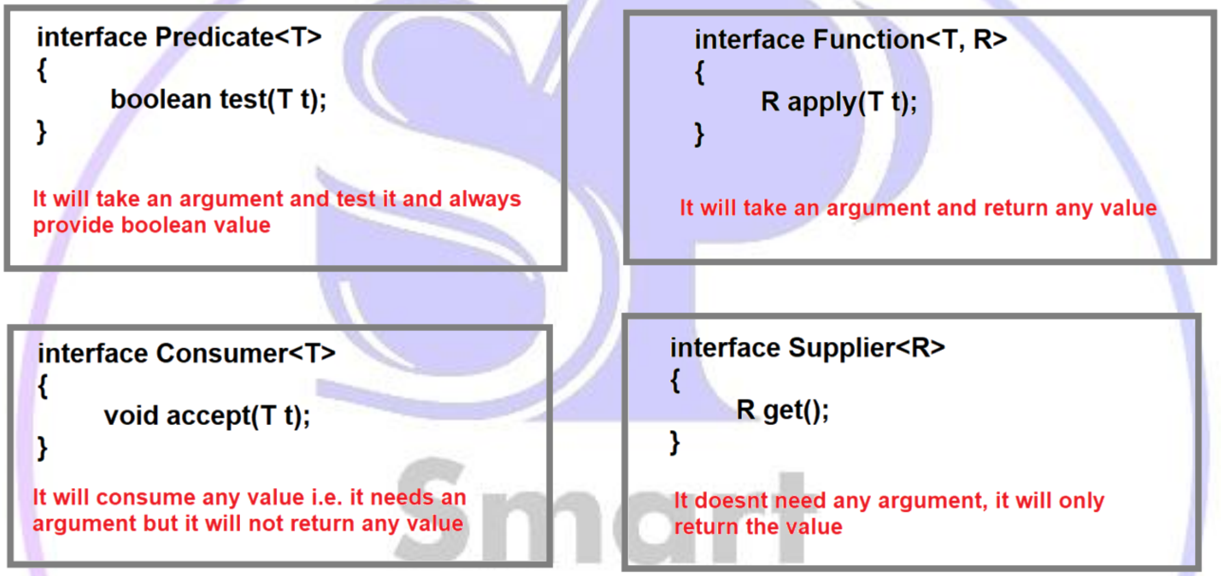
```
public interface Consumer<T>

{

        void accept(T t);

        //default methods..

}
```

=> Supplier :-

- ➔ Supplier is a functional interface which is present in java.util.function package
- ➔ Syntax :

```
public interface Supplier<R>
{
        R get();
}
```

=> What is difference between Predicate, Function, Consumer & Supplier



```
interface Predicate<T>
{
        boolean test(T t);
}
```

It will take an argument and test it and always provide boolean value

```
interface Function<T, R>
{
        R apply(T t);
}
```

It will take an argument and return any value

```
interface Consumer<T>
{
        void accept(T t);
}
```

It will consume any value i.e. it needs an argument but it will not return any value

```
interface Supplier<R>
{
        R get();
}
```

It doesnt need any argument, it will only return the value

Company Links & Contacts

Company Name: Smart Programming (+91 62838-30308)

Address : Chandigarh & Mohali (Punjab), India

Websites: https://www.smartprogramming.in/

https://courses.smartprogramming.in

Android App: https://play.google.com/store/apps/details?id=com.smartprogramming

YouTube Channel: https://www.youtube.com/c/SmartProgramming

Smart Programming : YouTube Channel

An investment in Knowledge pays the best interest….

Smart Programming
We Educate - We Develop

+91 62838-30308
Call us to Learn
Latest Technologies

City : Mohali (Punjab),
& Chandigarh
(India)

WEBSITE : http://www.smartprogramming.in

BUY COURSES ON : https://courses.smartprogramming.in

YOUTUBE CHANNEL : Smart Programming (https://www.youtube.com/c/SmartProgramming)

ANDROID APP : Smart Programming
(https://play.google.com/store/apps/details?id=com.smartprogramming)

https://www.facebook.com/smartprogramming.india

https://www.instagram.com/smart_programming

Java
android
python
Data Science
Machine Learning
Artificial Intelligence
M E A N
Full Stack
BIG DATA
hadoop

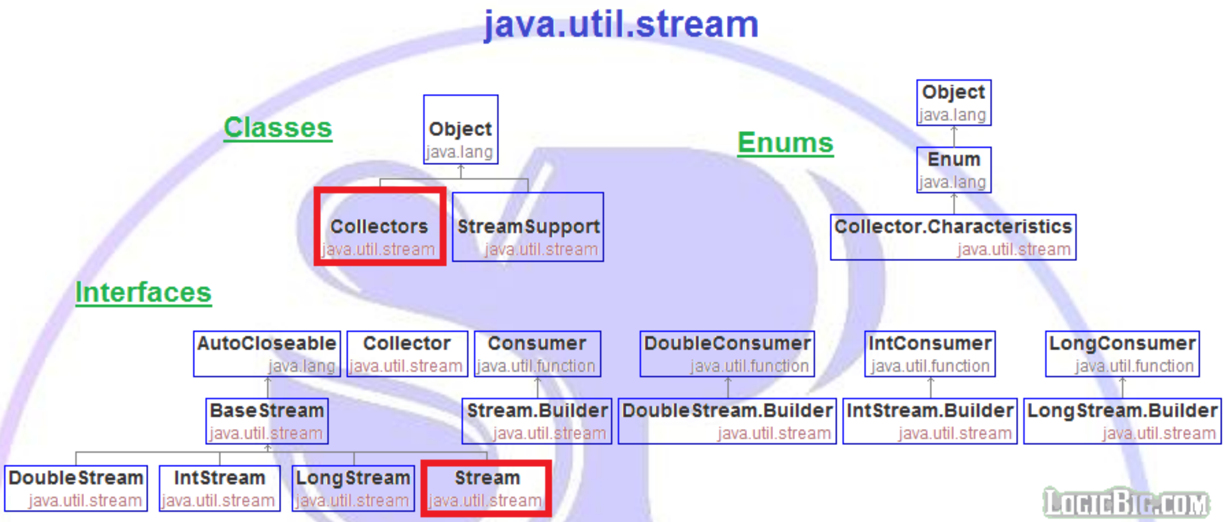
Stream API in Java

=> Stream API :-

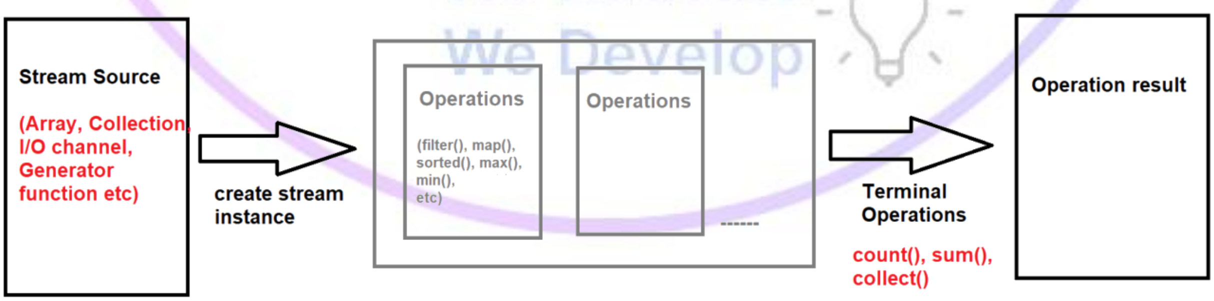
- ➔ Stream API is related to "Collection Framework" or "Group of Objects"
- ➔ Note : Stream API is not related to Java IO Stream, it is totally different from IO stream
- ➔ Stream API was introduced in Java SE 8 version
- ➔ Use of Stream API :-

1. It is used to process or to perform bulk operations on collection objects or group of objects
2. It decreases the number of lines of code

➔ Hierarchy of Stream API :-


java.util.stream
Classes
Object java.lang
Collectors java.util.stream
StreamSupport java.util.stream
Enums
Object java.lang
Enum java.lang
Collector.Characteristics java.util.stream
Interfaces
AutoCloseable java.lang
Collector java.util.stream
Consumer java.util.function
DoubleConsumer java.util.function
IntConsumer java.util.function
LongConsumer java.util.function
BaseStream java.util.stream
Stream.Builder java.util.stream
DoubleStream.Builder java.util.stream
IntStream.Builder java.util.stream
LongStream.Builder java.util.stream
DoubleStream java.util.stream
IntStream java.util.stream
LongStream java.util.stream
Stream java.util.stream
LOGICBIG.COM

➔ Stream API Operations Lifecycle :-


Stream Source
(Array, Collection, I/O channel, Generator function etc)
create stream instance
Operations
(filter(), map(), sorted(), max(), min(), etc)
Operations

Terminal Operations
count(), sum(), collect()
Operation result

1. Get stream from the source and create stream instance
2. Perform operations(0,1 or many operations) such as filter(), sort(), map() etc and transform it into another stream
3. Perform terminal operation such as count(), sum() etc and produces the result

---

=> Stream Interface :-

-> Stream is an interface which is present in java.util.stream package

-> Syntax :

```
public interface Stream
{
        forEach();
        filter();
        map();
        collect();

        sorted();
        min();
        max();
        count();

        //some static methods
        of() { - }
        empty() { - }
        builder() { - }
}
```

=> Task :-

1. WAP to print all the numbers which are greater than 30
2. WAP to print all the string values whose length is greater than 5
    a. (deepak, deepesh, amit, rahul, ankit) -> deepak, deepesh

2. WAP to print all the names whose name starts with d

3. WAP to get the first character of each name
4. WAP to replace the a charcter with z character

5. WAP to sort an arraylist containg integer values

6. WAP to get Minimum and Maximum element in an ArrayList

7. WAP to print the sum of all the even numbers in an ArrayList

---

Company Links & Contacts

Company Name: Smart Programming (+91 62838-30308)

Address : Chandigarh & Mohali (Punjab), India

Websites: https://www.smartprogramming.in/

https://courses.smartprogramming.in

Android App: https://play.google.com/store/apps/details?id=com.smartprogramming

YouTube Channel: https://www.youtube.com/c/SmartProgramming

Smart Programming : YouTube Channel

An investment in Knowledge pays the best interest….

Smart Programming
We Educate - We Develop

+91 62838-30308
Call us to Learn Latest Technologies

City : Mohali (Punjab), & Chandigarh (India)

WEBSITE : http://www.smartprogramming.in

BUY COURSES ON : https://courses.smartprogramming.in

YOUTUBE CHANNEL : Smart Programming (https://www.youtube.com/c/SmartProgramming)

ANDROID APP : Smart Programming (https://play.google.com/store/apps/details?id=com.smartprogramming)

https://www.facebook.com/smartprogramming.india

https://www.instagram.com/smart_programming

Java
android
python
Data Science
Machine Learning
Artificial Intelligence
M E A N
Full Stack
BIG DATA
hadoop

Date & Time API in Java

=> Date & Time API :-

- ➔ Date and Time API was in previous versions also but new Date-Time API was introduced in Java SE 8 version
- ➔ The packages for Date-Time API are :-

1. java.util (old - till JDK 7)
    - a. java.util.Date
    - b. java.util.Calendar
    - c. java.util.TimeZone
2. java.sql
    - a. java.sql.Date
    - b. java.sql.Time
    - c. java.sql.Timestamp
3. java.text
    - a. java.text.SimpleDateFormat
    - b. java.text.DateFormat
4. java.time (new - JDK 8)
    - a. java.time.LocalDate
    - b. java.time.LocalTime
    - c. java.time.LocalDateTime
    - d. java.time.ZoneId
    - e. java.time.Clock

f. java.time.Year
g. java.time.YearMonth
h. java.time.Period
i. java.time.Duration
j. java.time.Instant
k. java.time.MonthDay
l. etc

=> What is difference between old and new Date API

1. In old Date-Time API, date was mutable

In new Date-Time API, date is not mutable (immutable)

2. In Old Date-Time API, SimpleDateFormat was not thread-safe
   In New Date-Time API, Formatter are thread-safe

3. In Old Date-Time API, we can get date and time by using only one class
   In New Date-Time API, there are also different classes for date and time

4. In Old Date-Time API, Year starts with 1900 and Months starts from 0
   New Date-Time API, years and months int values were modified

Company Links & Contacts

Company Name: Smart Programming (+91 62838-30308)

Address : Chandigarh & Mohali (Punjab), India

Websites: https://www.smartprogramming.in/

https://courses.smartprogramming.in

Android App:
https://play.google.com/store/apps/details?id=com.smartprogramming

YouTube Channel:
https://www.youtube.com/c/SmartProgramming

Smart Programming : YouTube Channel

An investment in Knowledge pays the best interest….

Smart Programming
We Educate - We Develop

+91 62838-30308
Call us to Learn
Latest Technologies

City : Mohali (Punjab),
& Chandigarh
(India)

WEBSITE : http://www.smartprogramming.in

BUY COURSES ON : https://courses.smartprogramming.in

YOUTUBE CHANNEL : Smart Programming (https://www.youtube.com/c/SmartProgramming)

ANDROID APP : Smart Programming
(https://play.google.com/store/apps/details?id=com.smartprogramming)

https://www.facebook.com/smartprogramming.india

https://www.instagram.com/smart_programming

Java
android
python
Data Science
Machine Learning
Artificial Intelligence
M E A N
Full Stack
BIG DATA
hadoop

SQL Queries

=> Database :-

- Database is a software which stores the data in an organized form.
- Database examples are : MySQL, Oracle, PostgreSQL, SQL Server etc

- Note : SQL is a query language which is used to manipulate database (SQL is not a database)

- Download : Download from "https://www.mysql.com/downloads/"

=> Type of SQL Commands :-

1. DDL (Data Defination Language) :- Create, Alter, Drop, Rename, Truncate, Comment
2. DML (Data Maniplulation Language) :- Insert, Delete, Update, Merge, Call
3. DQL (Data Query Language) :- Select
4. TCC (Transaction Control Commands) :- Commit, Rollback, Savepoint

5. DCL (Data Control Language) :- Alter password, Grant Access, Revoke Access etc
6. DAC (Data Administration Commands) :- Start Audit, Stop Audit etc

=> MySQL Commands :-

1. CREATE DATABASE datebase_name; (it will create new database)
2. SHOW DATABASES; (it will display all the databases)
3. DROP DATABASE database_name; (it will delete the provided database)
4. USE database_name; (it will select the provided database)
5. CREATE TABLE table_name(column_name data-type(size), column_name data-type(size), -, -); (it will create table with provided column names)
6. SHOW TABLES; (it will display all the tables in selected database)
7. DESC tablename; (it will display the table details)
8. ALTER TABLE tablename ADD column_name data-type(size); (it will add new column in the table)

9. ALTER TABLE tablename MODIFY COLUMN column_name data-type(size); (it will modify the column)
10. ALTER TABLE tablename DROP COLUMN column_name; (it will delete the provided column)
11. DROP TABLE tablename; (it will delete the table)

12. INSERT INTO tablename VALUES('-', '-', '-', '-', '-');
13. INSERT INTO tablename(column_name, column_name, column_name) VALUES('-', '-', '-');
14. SELECT * FROM tablename;
15. SELECT column_name, column_name FROM tablename;
16. SELECT * FROM tablename WHERE column_name='value';

17. SELECT * FROM tablename WHERE column_name='value' AND/OR column_name='value';
18. SELECT column_name, column_name FROM tablename WHERE column_name='value' AND/OR column_name='value';

19. UPDATE tablename SET column_name='value', column_name='value' WHERE column_name='value';
20. DELETE FROM tablename WHERE column_name='value';

Company Links & Contacts

Company Name: Smart Programming (+91 62838-30308)

Address : Chandigarh & Mohali (Punjab), India

Websites: https://www.smartprogramming.in/

https://courses.smartprogramming.in

Android App:
https://play.google.com/store/apps/details?id=com.smartprogramming

YouTube Channel:
https://www.youtube.com/c/SmartProgramming